



Mellanox WinOF VPI Documentation

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

© Copyright 2019. Mellanox Technologies Ltd. All Rights Reserved.

Mellanox® and the Mellanox logo are registered trademarks of Mellanox Technologies, Ltd.

Additional trademarks used in this document are listed under <http://www.mellanox.com/page/trademarks>.

All other trademarks are property of their respective owners.

Table of Contents

1	Release Notes Rev 5.50.52000	11
1.1	WinOF VPI Package Contents	11
1.2	Supported Operating System Versions	11
1.3	Hardware and Software Requirements	13
1.4	Certifications	13
1.5	Supported Network Adapter Cards	14
1.5.1	Firmware Versions	14
1.5.2	MFT Version	14
1.6	RoCE Modes Matrix	15
1.7	Changes and New Features	15
1.7.1	Beta Features	15
1.7.2	Unsupported Functionalities/Features	16
1.8	Known Issues	17
1.9	Bug Fixes History	21
2	Introduction	42
2.1	Supplied Packages	42
2.2	WinOF Set of Documentation	43
2.3	Windows MPI (MS-MPI)	43
3	Installation	44
3.1	Hardware and Software Requirements	44
3.2	Downloading Mellanox WinOF Driver	44
3.3	Installing Mellanox WinOF Driver	45
3.3.1	Attended Installation	45
3.3.2	Unattended Installation	49
3.4	Uninstalling Mellanox WinOF Driver	50
3.4.1	Attended Uninstallation	50
3.4.2	Unattended Uninstallation	51
3.5	Installation Results	51
3.6	Extracting Files without Running Installation	52
3.7	Upgrading Firmware & WinOF Driver	54
3.7.1	Firmware Upgrade	54
3.7.2	Upgrading Mellanox WinOF Driver	54
3.8	Bootting Windows from an iSCSI Target or PXE	55
3.8.1	Configuring the WDS, DHCP and iSCSI Servers	55
3.8.1.1	Configuring the WDS Server	55
3.8.1.2	Configuring iSCSI Target	55

3.8.1.3	Configuring the DHCP Server	55
3.8.2	Configuring the Client Machine	56
3.8.3	Installing the Operating System	56
4	Features Overview and Configuration	59
4.1	Ethernet Network.....	59
4.1.1	Port Configuration	59
4.1.1.1	Auto Sensing	59
4.1.1.2	Port Protocol Configuration	60
4.1.2	Assigning Port IP After Installation	61
4.1.3	56GbE Link Speed	63
4.1.4	RDMA over Converged Ethernet (RoCE).....	64
4.1.4.1	RoCE Configuration	65
4.1.4.2	RoCEv2	66
4.1.4.3	RoCE v2 UDP Port.....	67
4.1.4.4	Configuring RoCE	68
4.1.5	Teaming and VLAN	71
4.1.5.1	Configuring a Port to Work with VLAN in Windows Server 2012 and Above	71
4.1.6	Command Line Based Teaming Configuration.....	71
4.1.6.1	Show Help	71
4.1.6.2	Show all Adapters (including already created teams)	72
4.1.6.3	Create an Empty Team.....	72
4.1.6.4	Attach Members to Team	73
4.1.6.5	Create a VLAN	73
4.1.6.6	Modify a Team	73
4.1.6.7	Modify a VLAN	73
4.1.6.8	ShowList (including the team created)	74
4.1.6.9	QueryTeam.....	74
4.1.6.10	QueryVlan	74
4.1.6.11	Remove VLAN.....	75
4.1.6.12	Remove an Entire Team.....	75
4.1.6.13	Show List Again (back to the beginning)	75
4.1.7	Header Data Split	75
4.1.8	Receive Segment Coalescing (RSC)	75
4.1.8.1	System Requirements	76
4.1.8.2	RSC Counters	76
4.1.9	Configuring Quality of Service (QoS)	77
4.1.9.1	System Requirements	77
4.1.9.2	QoS Configuration.....	77

4.1.9.3	Enhanced Transmission Selection	80
4.1.10	Configuring the Ethernet Driver	81
4.1.11	Differentiated Services Code Point (DSCP)	82
4.1.11.1	System Requirements	82
4.1.11.2	Setting the DSCP in the IP Header	82
4.1.11.3	Configuring Quality of Service for TCP and RDMA Traffic	83
4.1.11.4	Configuring DSCP to Control PFC for TCP Traffic	83
4.1.11.5	Configuring DSCP to Control ETS for TCP Traffic	83
4.1.11.6	Configuring DSCP to Control PFC for RDMA Traffic	84
4.1.11.7	Registry Settings	84
4.1.11.8	DSCP Sanity Testing	86
4.1.12	Lossless TCP	86
4.1.12.1	System Requirements	86
4.1.12.2	Using Lossless TCP	86
4.1.12.3	Known Limitations	87
4.1.12.4	System Requirements	88
4.1.12.5	Enabling/Disabling Lossless TCP	88
4.1.12.6	Monitoring Lossless TCP State	88
4.1.13	Receive Side Scaling (RSS).....	89
4.1.13.1	System Requirements	89
4.1.13.2	Using RSS	89
4.1.14	Ignore Frame Check Sequence (FCS) Errors	90
4.1.15	VXLAN	90
4.1.16	Threaded DPC	90
4.1.16.1	Registry Configuration	90
4.2	InfiniBand Network	90
4.2.1	Port Configuration	91
4.2.2	Assigning Port IP After Installation	91
4.2.3	Receive Side Scaling (RSS).....	91
4.2.4	OpenSM - Subnet Manager	91
4.2.4.1	Modifying IPoIB Configuration	92
4.2.4.2	Displaying Adapter Related Information	92
4.2.5	Modifying IPoIB Configuration	94
4.2.6	Displaying Adapter Related Information	94
4.2.7	Multiple Interfaces over Non-Default PKeys Support	95
4.2.7.1	System Requirements	95
4.2.7.2	Using Multiple Interfaces over non-default PKeys	95
4.2.8	Teaming	96

4.2.8.1	System Requirements	97
4.2.8.2	Adapter Teaming	97
4.2.8.3	Creating a Team	97
4.3	Management	99
4.3.1	PowerShell Configuration	99
4.4	Storage Protocols	100
4.4.1	Deploying Windows Server 2012 and Above with SMB Direct	100
4.4.1.1	System Requirements	101
4.4.2	SMB Configuration Verification.....	101
4.4.2.1	Verifying Network Adapter Configuration.....	101
4.4.2.2	Verifying SMB Configuration	101
4.4.2.3	Verifying SMB Connection.....	101
4.4.2.4	Verifying SMB Events that Confirm RDMA Connection	102
4.5	Virtualization	102
4.5.1	Hyper-V with VMQ.....	102
4.5.1.1	System Requirements	102
4.5.1.2	Using Hyper-V with VMQ	102
4.5.2	Network Virtualization using Generic Routing Encapsulation (NVGRE)	103
4.5.2.1	Enabling/Disabling NVGRE Offloading	103
4.5.3	Single Root I/O Virtualization (SR-IOV).....	103
4.5.3.1	SR-IOV Ethernet over Hyper-V	104
4.5.3.2	SR-IOV InfiniBand over KVM	104
4.5.3.3	Configuring SR-IOV Host Machines.....	105
4.5.3.4	Configuring Mellanox Network Adapter for SR-IOV	111
4.5.3.5	Configuring Operating Systems	115
4.5.4	Virtual Machine Multiple Queue (VMMQ).....	120
4.5.4.1	System Requirements	120
4.5.4.2	Enabling/Disabling VMMQ	120
4.5.4.3	Controlling the Number of Queues Allocated for a vPort	121
4.5.5	PacketDirect Provider Interface.....	122
4.5.5.1	System Requirements	122
4.5.5.2	Using PacketDirect for VM	122
4.6	Configuration Using Registry Keys.....	125
4.6.1	Finding the Index Value of the HCA	126
4.6.2	Finding the Index Value of the Network Interface.....	126
4.6.3	Basic Registry Keys	127
4.6.4	Off-load Registry Keys	129
4.6.5	Performance Registry Keys.....	132

4.6.6	Ethernet Registry Keys.....	136
4.6.6.1	Flow Control Options.....	138
4.6.6.2	VMQ Options.....	138
4.6.7	IPoB Registry Keys.....	139
4.6.8	General Registry Values.....	141
4.6.9	MLX BUS Registry Keys	142
4.6.9.1	SR-IOV Registry Keys.....	142
4.6.9.2	RoCE Options	143
4.6.9.3	General Registry Keys	143
4.7	Dump Me Now (DMN).....	143
4.7.1	DMN Triggers and APIs.....	144
4.7.2	Dumps and Incident Folders	144
4.7.3	Cyclic DMN Mechanism	145
4.7.4	Configuration	145
4.7.4.1	DMN-IOV Configuration	146
4.7.5	Event Logs	146
4.8	Software Development Kit (SDK).....	147
4.8.1	Network Direct Interface.....	147
4.8.2	Win-Linux nd_rping Test.....	147
4.8.2.1	Test Running	147
4.9	Performance Tuning and Counters	148
4.9.1	General Performance Optimization and Tuning	149
4.9.1.1	Mellanox Specific Extensions to the ND Interface	149
4.9.1.2	Registry Tuning	150
4.9.1.3	Enable RSS.....	150
4.9.1.4	Tuning the IPoB Network Adapter	150
4.9.1.5	Tuning the Ethernet Network Adapter	151
4.9.1.6	Performance Tuning Tool Application	152
4.9.1.7	SR-IOV Tuning	155
4.9.1.8	Improving Live Migration	155
4.9.2	Application Specific Optimization and Tuning	155
4.9.2.1	Ethernet Performance Tuning	155
4.9.3	Tunable Performance Parameters	156
4.9.4	Adapter Proprietary Performance Counters.....	157
4.9.4.1	Proprietary Mellanox Adapter Traffic Counters.....	158
4.9.4.2	Proprietary Mellanox Adapter Diagnostics Counters.....	159
4.9.4.3	Proprietary Mellanox QoS Counters.....	162
4.9.4.4	RSS Monitoring	163

4.9.5	Device Proprietary Counters	169
4.9.5.1	Mellanox Proprietary WinOF Bus Counters	170
4.10	Resiliency	172
4.10.1	Device Self-Healing	172
4.10.1.1	Health-Checker Mechanism	173
4.10.1.2	Configuration	175
4.10.1.3	Logging	175
4.11	RDMA Features	176
4.11.1	ND2 Provider Control	176
4.11.1.1	IND2ProviderControl Interface	176
4.11.2	Disabling Mellanox Device while NetworkDirect Applications are Running	179
5	Utilities	180
5.1	Snapshot Tool.....	180
5.2	part_man - Virtual IPoIB Port Creation Utility	180
5.3	InfiniBand Fabric Diagnostic Utilities	182
5.3.1	Utilities Usage: Common Configuration, Interface and Addressing	182
5.4	Fabric Performance Utilities	185
5.5	mlxtool	186
5.5.1	mlxtool Help.....	187
5.5.2	dbg sub-tool.....	187
5.5.2.1	nonrss-capture	187
5.5.2.2	sw-reset	188
5.5.2.3	mstdump.....	188
5.5.2.4	oid-stats	189
5.5.2.5	cmd-stats	189
5.5.2.6	pkeys	190
5.5.2.7	Resources	190
5.5.2.8	Ipoib-ep	192
5.5.2.9	Get-state.....	192
5.5.2.10	Restart.....	193
5.5.2.11	Port Diagnostic Database Register	193
5.5.3	show	194
5.5.3.1	show packet-filter	194
5.5.3.2	show qos	194
5.5.3.3	show nd	195
5.5.3.4	show port list	195
5.5.3.5	show device list	195
5.5.3.6	Show vxlan	195

5.5.3.7	Show selfhealing	196
5.5.3.8	regkeys	197
5.5.3.9	show driverParams	199
5.5.3.10	show perfstats	200
5.5.4	modify Tool	202
5.5.4.1	Modify Traffic Classes Bandwidth (BW) Limit Configuration Tool	202
6	Troubleshooting	204
6.1	RDMA Related Troubleshooting	204
6.2	Installation Related Troubleshooting	204
6.2.1	Installation Error Codes and Troubleshooting	204
6.2.1.1	Setup Return Codes	205
6.2.1.2	Firmware Burning Warning Codes	205
6.2.1.3	Restore Configuration Warnings	205
6.3	InfiniBand Related Troubleshooting	206
6.4	Ethernet Related Troubleshooting	206
6.5	Performance Related Troubleshooting	207
6.5.1	General Diagnostic	208
6.6	Virtualization Related Troubleshooting	209
6.7	Reported Driver Events	210
6.8	Extracting WPP Traces	211
6.9	State Dumping	211
7	Appendix: Windows MPI (MS-MPI)	213
7.1	System Requirements	213
7.2	Running MPI	213
7.3	Directing MSMPI Traffic	213
7.4	Running MSMPI on the Desired Priority	213
7.5	Configuring MPI	214
7.5.1	PFC Example	214
7.5.2	Running MPI Command Examples	215
8	Document Conventions and Related Documents	216
8.1	Document Conventions	216
8.2	Abbreviations and Acronyms	216
8.3	Related Documentation	217
9	User Manual Revision History	219
10	Release Notes Change Log History	221
11	API Change Log History	248

Overview

Mellanox WinOF is the Windows driver for ConnectX-3 and ConnectX-3 Pro network adapter cards. The documentation describes WinOF features, performance, diagnostic tools, content and configuration. Additionally, this document provides information on various performance tools supplied with this version.

The documentation here relates to WinOF:

- [Release Notes Rev 5.50.52000](#)
- [User Manual](#)

Software Download

To download product software, please refer to the [Mellanox OFED for Windows](#) product page.

Document Revision History

A list of the changes made to the user manual are provided in [User Manual Revision History](#).

1 Release Notes Rev 5.50.52000

These are the release notes for the release of Mellanox WinOF VPI Drivers for Windows, supporting Mellanox ConnectX®-3 and ConnectX®-3 Pro network adapters.

Mellanox WinOF is composed of several software modules that contain InfiniBand and Ethernet drivers and utilities for ConnectX®-3 and ConnectX®-3 Pro adapter cards.

⚠ Windows Server 2012, Windows Server 2012 R2 and Windows Server 2016 include WinOF Inbox drivers which are a subset of the full WinOF VPI software package. As the Inbox drivers do not provide the full performance and functionality available with the WinOF VPI package, we recommend installing the full and latest WinOF VPI package.

- [WinOF VPI Package Contents](#)
- [Supported Operating System Versions](#)
- [Hardware and Software Requirements](#)
- [Certifications](#)
- [Supported Network Adapter Cards](#)
- [RoCE Modes Matrix](#)
- [Changes and New Features](#)
- [Known Issues](#)
- [Bug Fixes History](#)

1.1 WinOF VPI Package Contents

The Mellanox WinOF driver package contains the following components:

- Core and ULPs:
 - IB HCA low-level drivers (mlx4)
 - Ethernet driver (ETH)
 - IP over InfiniBand (IPoIB)
 - Network Direct (ND)
 - Network Direct Kernel (NDK) Provider Interface
 - MUX driver for IPoIB and Client OSes
- Utilities:
 - OpenSM: InfiniBand Subnet Manager is provided as a sample code. The sample code is intended to allow users to test or bring-up the InfiniBand fabric without a management console / switch (to get started). For cluster production environments, Mellanox's recommendation is to use a Managed Switch or the UFM-SDN Appliance.
 - Low level performance tools
 - mlxtool
- InfiniBand Diagnostics tools
- CIM, PowerShell, and WMI support (Supported in Windows Server 2012 and above, and Windows Client 8.1 and above.)
- Software Development Kit (SDK)
- Documentation

1.2 Supported Operating System Versions

The following describes the supported operating systems and their roles in a virtualization environment.

Supported Host OS	Supported Guest OS
Virtualization Mode: None	

Supported Host OS	Supported Guest OS
Windows Server 2012	N/A
Windows Server 2012 R2	N/A
Windows Server 2016	N/A
Windows Server 2019	N/A
Windows 8.1 Client (64 bit only)	N/A
Windows 10 Client 1607 (64 bit only)	N/A
Windows 10 Client 1709 (64 bit only)	N/A
Windows 10 Client 1803 (64 bit only)	N/A
Windows 10 Client 1809 (64 bit only)	N/A
Virtualization Mode: VMQ	
Windows Server 2012	Any supported guest OS for Hyper-V
Windows Server 2012 R2	Any supported guest OS for Hyper-V
Windows Server 2016	Any supported guest OS for Hyper-V
Windows Server 2019	Any supported guest OS for Hyper-V
Virtualization Mode: VMMQ	
Windows Server 2016	Any supported guest OS for Hyper-V
Windows Server 2019	Any supported guest OS for Hyper-V
Virtualization Mode: Hyper-V (SR-IOV)	
Windows Server 2012 R2	Windows Server 2012
	Windows Server 2012 R2
	Windows Client 8.1
	Windows Client 10 1809
Windows Server 2016	Windows Server 2012

Supported Host OS	Supported Guest OS
	Windows Server 2012 R2
	Windows Server 2016
	Windows Client 8.1
	Windows Client 10 1809
Windows Server 2019	Windows Server 2019
	Windows Server 2016
SR-IOV InfiniBand KVM	Windows Server 2012
	Windows Server 2012 R2
	Windows Server 2016
SR-IOV Ethernet KVM	Windows Server 2012
	Windows Server 2012 R2
	Windows Server 2016

1.3 Hardware and Software Requirements

The following are the hardware and software requirements of WinOF driver:

- The maximum number of supported CPUs is:
 - up to 252 logical processors when Hyper-threading is enabled
 - up to 126 logical processors when Hyper-threading is disabled
- Administrator privileges on your machine(s)
- Disk space: Minimum 100MB

1.4 Certifications

The following describes the driver's certification status per operating system.

Operating System	Logo Certification	SDDC Premium Certification
Windows 8.1 Client	N/A	N/A
Windows 10 Client 1607	N/A	N/A
Windows 10 Client 1809	N/A	N/A
Windows Server 2012	N/A	N/A


Operating System	Logo Certification	SDDC Premium Certification
Windows Server 2012 R2	N/A	N/A
Windows Server 2016	N/A	N/A
Windows Server 2019	N/A	N/A

1.5 Supported Network Adapter Cards

Mellanox WinOF driver supports the following Mellanox network adapter cards:

NICs	Supported Protocol	Supported Link Speed
ConnectX®-3 Pro	InfiniBand (IB)	SDR, DDR, QDR, FDR10, FDR
	Ethernet	10, 40 and 56 Gb/s
ConnectX®-3	InfiniBand (IB) ^a	SDR, DDR, QDR, FDR10, FDR
	Ethernet ^a	10, 40 and 56 Gb/s

Note a. This version is supported by the ConnectX®-3 adapter card, but is officially qualified for ConnectX®-3 Pro only.

 The speeds listed in the table above are according to the speeds supported by the device.

1.5.1 Firmware Versions

Mellanox WinOF driver provides the following firmware for Mellanox NICs:

NICs	Recommended Firmware Rev.	Additional Firmware Rev. Supported
ConnectX®-3 Pro / ConnectX®-3 Pro EN	2.42.5000	2.40.7000
ConnectX®-3 / ConnectX®-3 EN	2.42.5000	2.40.7000

1.5.2 MFT Version

Mellanox WinOF driver provides the following MFT version:

	Supported Version
MFT for Windows	4.11.0

1.6 RoCE Modes Matrix

The following is the RoCE modes matrix.

Software Stack / Inbox Distribution	RoCE MAC Based (Layer 2) Supported as of Version	RoCE IP Based (Layer 2) Supported as of Version	RoCE v2 (Layer 3) Supported as of Version
Mellanox WinOF	3.2 (Default)	4.80 (Requires additional configuration)	<ul style="list-style-type: none"> 4.70 (Requires additional configuration) 5.22 (Default)
Inbox Windows Server 2012 / Inbox Windows Server 2012 R2	Supported (Default)	Not supported	Not supported
Inbox Windows Server 2016	Supported	Supported	Supported (Default)

⚠ As of WinOF 5.22, RoCE v2 is the default RoCE mode.

RoCE v2 UDP Port Matrix

WinOF Versions	RoCE v2 UDP Port
4.70 - 5.00	1021
5.10 and above	4791

For further information, please refer to section [RoCEv2 UDP Port](#) in the User Manual.

1.7 Changes and New Features

⚠ This package version is 5.50.52000. The package contains the following versions of components:

- Bus, eth, IPoIB and MUX drivers version is 5.50.14688
- The CIM Provider version is 5.50.14688

Category	Description
Bug Fixes	See Bug Fixes History .

1.7.1 Beta Features

Category	Description
ibdump	ibdump is currently at beta level.

Category	Description
IPoIB	IPv6 support of IPoIB in an SR-IOV guest OS over KVM.
	IPoIB teaming support is at beta level and is supported only on native machines (and not in HyperV or SR-IOV).

1.7.2 Unsupported Functionalities/Features

The following are the unsupported functionalities/features in WinOF:

Functionality/Feature	Not Supported as of WinOF Version
Software RSC for tunneled traffic	WinOF v5.50
Wake-On-Lan	
RDMA in the Guest OSes	
ND over virtual switch attached to IPoIB port	
Memory Translation Table (MTT) Optimization	
RDMA over VM when in SR IOV mode	WinOF 5.35
IBVerbs	WinOF 5.22
WinVerbs	
IBAL performance tools (ib* ibv*)	WinOF 5.10
IBAL ND provider	
IPoIB team configuration through PowerShell	WinOF 4.90
IPv6 over IPoIB team ports	
VMQ over IPoIB team ports	
ConnectX®-2 adapter cards	
ND over WinVerbs provider	WinOF 4.52
SRP	

1.8 Known Issues

The table below provides a list of known bugs and limitations in regards to this release of WinOF.

For a list of old Known Issues, please see the [WinOF_Archived_Known_Issues](#) file.

Internal Ref.	Issue
1732070	Description: When creating a team on Windows Client 10 in one of the following versions (1803 or 1809), a waiting time of a few seconds is required after running the 'vlan_config attach...' command. If the waiting period is skipped, an error message is received when running the PS command 'Get-NetAdapter'. This message can be safely ignored as it does not affect the team's creation or interferes with the correct behavior of the team.
	Workaround: N/A
	Keywords: Teaming
	Detected in version: 5.50.52000
1371587	Description: When in IPoIB mode, changing the receive buffers size using a registry key and restarting the driver while the RDMA traffic is running may result in a command failure.
	Workaround: N/A
	Keywords: IPoIB
	Detected in version: 5.50
1368272	Description: The "Disable-NetAdapterRDMA" command disables the NDK activity only, ND activity is not affected by it.
	Workaround: N/A
	Keywords: NDK, ND
	Detected in version: 5.50
1297888	Description: On Windows 2016 Client: The SR-IOV virtual adapters' counters do not count the data of the Virtual Function. The counters will not rise when the Virtual Function's counters rise neither in the performance monitor nor in the task manager.
	Workaround: N/A
	Keywords: SR-IOV, VF Counters
	Detected in version: 5.50

Internal Ref.	Issue
1139573	Description: The maximum value of the *NumRSSQueues registry key is 64.
	Workaround: N/A
	Keywords: *NumRSSQueues registry key
	Detected in version: 5.50
1048287	Description: When setting the MaxCMRetries value to 1, the REQ, REP, or DREQ messages are not sent.
	Workaround: N/A
	Keywords: MaxCMRetries
	Detected in version: 5.50
1158964	Description: The driver will ask for a reboot when a network interface gets disconnected while loading the VMs on hyper-V.
	Workaround: Disable and enable the driver manually, without rebooting. Note, you may have to perform this action twice to resolve the issue.
	Keywords: VMs, hyper-V, reboot
	Detected in version: 5.50
1158964	Description: When a network interface gets disconnected while loading VMs on hyper-V, the driver will ask for reboot.
	Workaround: Disable and enable the driver manually, without rebooting. You may have to perform this action twice to resolve the issue.
	Keywords: VMs, hyper-V, reboot
	Detected in version: 5.50
1130716	Description: The mlxtool show devices command is not supported for VFs.
	Workaround: N/A
	Keywords: mlxtool show devices; VFs

Internal Ref.	Issue
	Detected in version: 5.50
1046418	Description: Switching between upgrading with INF and upgrading with the MSI package can end up with a different version of the Ethernet/IPoIB driver and the bus driver.
	Workaround: Use the same upgrade methodology in all upgrades. Following the upgrade, update the device driver via the device manager.
	Keywords: Installation, INF only installation.
	Detected in version: 5.40.54000
928517	Description: When configuring certain device settings to invalid values in the driver's Advanced Properties tab, a random number is used as the actual value, instead of the default number.
	Workaround: N/A
	Keywords: device settings, advanced properties
	Discovered in Release: 5.40.54000
1034664	Description: In case the network adapter of the remote ND connection was reset, the local peer does not receive a disconnect notification.
	Workaround: To make sure that the connection is still valid, post a send request on the QP and check its completion status.
	Keywords: ND, disconnect notification
	Discovered in Release: 5.40.54000
1036061	<p>Description: When attempting to allocate several VFs to several VMs, in some cases the VF will not be allocated to the VM, and the VM will continue to work on the synthetic path.</p> <p>The user can identify the issue by running the Get-Netadaptervport command, and making sure that the number of MAC addresses is equal to the number of VFs.</p> <p>The issue occurs due to a bug in the Windows NetVSC co-installer, and has been fixed in a QFE. It occurs only in Windows Server 2012 R2.</p>
	<p>Workaround: Run the following commands in the problematic VM:</p> <pre>RUNDLL32.exe pnpclean.dll,RunDLL_PnpClean /devices /maxclean netcfg -c p -i netvsc_vfpp -l c:\windows\inf\wnetvsc_vfpp.inf</pre>

Internal Ref.	Issue
	Keywords: SR-IOV
	Discovered in Release: 5.40.54000
1020140	Description: When using Windows Server 2012 R2 or Windows Server 2016, in some cases a system crash may occur due to an OS bug when disabling the NDK. The issue will be resolved in the next version of Windows Server.
	Workaround: N/A
	Keywords: NDK, BSOD
	Discovered in Release: 5.40.54000
981757	Description: After driver upgrade, The ND state might be invalid, and the following event might appear in the event viewer: "Ndfldr: ND is in invalid state as a result of a mismatch between the ndfldr.sys driver version and mlx4_bus.sys driver version."
	Workaround: Close all applications that use ND before upgrading the driver or upgrade the driver first, close all applications that use ND, and restart the bus driver.
	Keywords: Installation, upgrade, ND
	Discovered in Release: 5.40.54000
1081254	Description: RDMA Activity counters are not support in IPoIB.
	Workaround: N/A
	Keywords: RDMA Activity counters, IPoIB
	Detected in version: 5.40
-	Description: When an Ethernet port is set to SR-IOV, InfiniBand is not supported on the second port.
	Workaround: N/A
	Keywords: SR-IOV, Ethernet, InfiniBand
	Detected in version: 5.40

Internal Ref.	Issue
1337021	Description: Received RDMA Activity Ack packets are counted as 20B instead of ~70B.
	Workaround: N/A
	Keywords: Counters
	Detected in version: 5.40

1.9 Bug Fixes History

The table below lists the bugs fixed in this release. For a list of old Bug Fixes, please see [WinOF_Archived_Bug_Fixes](#) file.

Internal Ref.	Issue
1484642	Description: Fixed an issue that resulted in a BSOD when changed the number of queues/CQ in VMMQ.
	Keywords: VMMQ, BSOD, RSS
	Discovered in Release: 5.50
	Fixed in Release: 5.50.52000
1572400	Description: Fixed an issue that resulted in system crash while updating the VPort in the error flow.
	Keywords: SR-IOV, VPort
	Discovered in Release: 5.50
	Fixed in Release: 5.50.52000
1424613	Description: Fixed a race condition that occurred when sending RDMA-send-messages between the storage nodes and the compute nodes which resulted in RDMA connectivity loss.
	Keywords: RDMA connectivity
	Discovered in Release: 5.50
	Fixed in Release: 5.50.52000
1467979	Description: Fixed an issue that prevented the NIC from enabling NVGRE or VXLAN although they were enabled by the user.
	Keywords: NVGRE, VXLAN
	Discovered in Release: 5.50
	Fixed in Release: 5.50.52000

Internal Ref.	Issue
1575666	Description: Fixed an issue which caused the systems_snapshot tool to hang when the ETL folder did not exist.
	Keywords: systems_snapshot
	Discovered in Release: 5.40
	Fixed in Release: 5.50.51000
1484642	Description: VMMQ: Fixed an issue that resulted in a BSOD due to an error when changing the number of queues/CQ.
	Keywords: VMMQ, BSOD, RSS
	Discovered in Release: 5.50
	Fixed in Release: 5.50.51000
1424259	Description: Fixed a race condition that occurred when simultaneously querying the Permon counters (the "Mellanox Adapter Traffic Counters" and the "Mellanox Adapter QoS Counters") and deleting the vPort OID, which resulted in BSOD.
	Keywords: Counters, VPorts
	Discovered in Release: 5.50
	Fixed in Release: 5.50.51000
1391835	Description: Fixed a rare issue that caused a deadlock between delete vPort and CheckForHang Routine.
	Keywords: Resiliency
	Discovered in Release: 5.50
	Fixed in Release: 5.50.51000
1365213	Description: Fixed an issue that occasionally caused the system to crash when the vNic was detached from the VM during heavy traffic when in VMQ\VMMQ mode.
	Keywords: VMQ\VMMQ mode
	Discovered in Release: 5.40.54000

Internal Ref.	Issue
	Fixed in Release: 5.50
1344714	Description: Fixed an issue where the RoCE connection failed as a result of an incorrect GID when the Universal/Local (U/L) bit in the MAC was set to 1.
	Keywords: RoCE
	Discovered in Release: 5.40.54000
	Fixed in Release: 5.50
1042285/ 1042206	Description: Fixed an issue that caused the mlxtool PDDR tool to provide some inaccurate information for Infiniband links.
	Keywords: mlxtool, PDDR, IPoIB
	Discovered in Release: 5.40.54000
	Fixed in Release: 5.50
1201166	Description: Disabled the option to stop the uninstall process once the driver uninstallation process started.
	Keywords: Driver uninstallation process
	Detected in version: 5.40.54000
	Fixed in Release: 5.50
1327365	Description: Fixed an issue that caused networks with new Subnet Managers (OpenSM 4.7.0 and up) to drop malformed multicast-join packets issued by the driver. The driver now constructs the multicast join request correctly.
	Keywords: OpenSM, multicast packets
	Detected in version: 5.35
	Fixed in Release: 5.50
1340828	Description: In case the DSCP values are lower than the max priority i.e: DSCP(4)->Prio(0) when mapping the DSCP to a certain priority, the priority's value will be set the same as the DSCP's value.

Internal Ref.	Issue
	<p>Keywords: DSCP counters</p> <p>Detected in version: 5.40</p> <p>Fixed in Release: 5.50</p>
1333906	<p>Description: Fixed an issue that caused the driver to hang when issuing an <code>OID_SRIOV_RESET_VF</code> request to reset a specified PCI Express (PCIe) Virtual Function (VF) due to a race between the resiliency flow and the FLR request.</p> <p>Keywords: Driver hang, PCI Express (PCIe) Virtual Function (VF)</p> <p>Discovered in Release: 5.40</p> <p>Fixed in Release: 5.50</p>
1252614	<p>Description: Fixed an issue that caused the driver to reset the adapter as a result of a false alarm of a stuck receive queue.</p> <p>Keywords: False alarm, Receive queue</p> <p>Discovered in Release: 5.40</p> <p>Fixed in Release: 5.50</p>
1266230	<p>Description: Fixed an issue that caused a Black Screen upon driver's removal due to extremely low memory conditions, when the memory allocations started to fail.</p> <p>Keywords: RDMA, NDK, Black Screen</p> <p>Discovered in Release: 5.40</p> <p>Fixed in Release: 5.50</p>
1261837	<p>Description: Fixed an issue that caused the binding to overrun the ND function INDEndpoint error status when it returned from the underlying functions. This resulted in wrong status display of the MR. The MR was displayed as registered when it was not, thus prevented the user from accessing it. This fix verifies that the user will receive the correct error status upon such scenario.</p> <p>Keywords: ND function INDEndpoint, MR</p> <p>Discovered in Release: 5.40</p>

Internal Ref.	Issue
	Fixed in Release: 5.50
1284856	Description: Fixed an issue that limited the number of MSI-X cores in Virtual Function to 8. Now the limited the number of MSI-X is 128 cores.
	Keywords: MSI-X vectors, VFs
	Discovered in Release: 5.40
	Fixed in Release: 5.50
1134253	Description: Fixed a BSOD that occurred on servers with more than 64 cores as the Tx traffic did not honor the Tx affinity implied by the TSS, when the number of potential RSS CPUs was greater than 64.
	Keywords: Tx traffic, RSS CPUs, TSS, BSOD
	Discovered in Release: 5.35
	Fixed in Release: 5.50
1078811	Description: When the mlxtool dbg resources command is executed, the FS_RULE quota number is displayed instead of the "Managed by PF" message.
	Keywords: mlxtool, dbg resources
	Detected in version: 5.30
	Fixed in Release: 5.50
1081576	Description: When setting the LogNumQp and LogNumRdmaRc registry settings to their maximum value, the WinOF bus driver fails to load.
	Keywords: LogNumQp, LogNumRdmaRc, driver load failure
	Detected in version: 5.40
	Fixed in Release: 5.50
1079136	Description: The "TX Ring Is Full Packets" perfmon counter is not functioning properly on IPoIB.
	Keywords: Perfmon counter, IPoIB

Internal Ref.	Issue
	Detected in version: 5.40
	Fixed in Release: 5.50
1038193	Description: When installing the driver over Windows 2012R2 inbox driver, the LogNumQP parameter remains in the registry. Thus, a number of QPs are limited to 64K instead of 512K (the driver's default).
	Keywords: Windows 2012R2 inbox, LogNumQP
	Discovered in Release: 5.40
	Fixed in Release: 5.50
1243974	Description: Fixed an issue that caused a system crash during driver startup when working in RSS mode.
	Keywords: RSS, system crash, driver startup
	Discovered in Release: 5.40
	Fixed in Release: 5.50
1170913	Description: Fixed an issue that caused a system crash when the interface connected to vSwitch was disabled and the operating system did not clean all VMQs.
	Keywords: IPoIB, VMQ
	Discovered in Release: 5.40
	Fixed in Release: 5.50
1213675	Description: Fixed a race in the Communication Manager that could cause a crash while attempting to obtain ND/NDK connection. This was an atypical issue that required an unusual timing of events.
	Keywords: CM, Communication Manager, connection, ND, NDK
	Discovered in Release: 5.40
	Fixed in Release: 5.50
1081160	Description: Fixed an issue that caused commands failure and protection domain violation when running the ND application.

Internal Ref.	Issue
	Keywords: ND application, commands failure, protection domain violation Discovered in Release: 5.40 Fixed in Release: 5.50
1170202	Description: Fixed an issue in the mlxtool, where the “mlxtool dbg ipoib-ep [<Interface Name>]” command reported partial results of the EndPoint list when there was a large number of endpoints. Keywords: mlxtool, dbg ipoib-ep Discovered in Release: 5.40 Fixed in Release: 5.50
1172093	Description: Fixed an issue that caused the VM to crash when restarting the PF drivers and their peers in the target machine. Keywords: PF, VF, driver restart, port down Discovered in Release: 5.40 Fixed in Release: 5.50
1182951	Description: Fixed an issue that caused a memory leak when RoCE was enabled. Keywords: Memory leak, RoCE Discovered in Release: 5.30 Fixed in Release: 5.50
1190576	Description: Fixed an issue that set a wrong value to the *ReceiveBuffers key when it was restored to default. Keywords: INF,*ReceiveBuffers Discovered in Release: 5.30 Fixed in Release: 5.50
1065413	Description: Fixed a crash that occurred when changing the Ethernet IP address while RDMA traffic was running.

Internal Ref.	Issue
	Keywords: Crash, IP address change, RDMA traffic
	Discovered in Release: 5.40
	Fixed in Release: 5.50
1070844	Description: Fixed a crash that occurred on IPoIB driver stack.
	Keywords: Crash, driver teardown, IPoIB
	Discovered in Release: 5.40
	Fixed in Release: 5.50
1082383	Description: Fixed a BSOD that occurred when a memory allocation failed upon driver startup.
	Keywords: Driver load, memory allocation failure, BSOD
	Discovered in Release: 5.40
	Fixed in Release: 5.50
1168384/ 1170019	Description: Fixed an issue where the connection port numbers did not increase sequentially when running nd_*_bw application with multiple QPs.
	Keywords: Connection port numbers, nd_*_bw
	Discovered in Release: 5.40
	Fixed in Release: 5.50
1131583	Description: Fixed an issue that caused a crash upon ND connection establishment.
	Keywords: CEP, Blue Screen crash
	Discovered in Release: 5.40
	Fixed in Release: 5.50
1134253	Description: Fixed an issue where the Tx traffic did not honor the Tx affinity implied by the TSS when the number of potential RSS CPUs was greater than 64.

Internal Ref.	Issue
	Keywords: TSS, RSS
	Discovered in Release: 5.40
	Fixed in Release: 5.50
1038193	Description: Fixed an issue that limited the number of QPs to 64K instead of 512K (the driver's default) when installing the driver over Windows 2012 R2 Inbox driver as the LogNumQP parameter remained in the registry/
	Keywords: Windows 2012R2 inbox, LogNumQP
	Discovered in Release: 5.40
	Fixed in Release: 5.50
936651	Description: Fixed an issue where removing a PKey that was a part of an IPoIB team interface disabled the team and the option to delete it.
	Keywords: IPoIB Pkeys, Team
	Discovered in Release: 5.40
	Fixed in Release: 5.50
1117581	Description: Added support for servers with more than 64 cores.
	Keywords: VMQ, SR-IOV
	Discovered in Release: 5.30
	Fixed in Release: 5.50
1037915	Description: Fixed a deadlock in which the driver could get into an error state in case resetting it and attempting to connect to it were performed simultaneously.
	Keywords: deadlock, driver reset
	Discovered in Release: 5.40
	Fixed in Release: 5.50

Internal Ref.	Issue
1077015	Description: Fixed an issue that could cause the Get-MlnxPCIDeviceSriovSetting command to display a wrong number of HCAs.
	Keywords: Get-MlnxPCIDeviceSriovSetting, HCA number
	Discovered in Release: 5.45
	Fixed in Release: 5.50
1078808	Description: Fixed an issue where the mlxtool dbg resources command failed to pull information about the last VF, and showed the PF as VF0.
	Keywords: mlxtool
	Discovered in Release: 5.40
	Fixed in Release: 5.50
1081045	Description: Fixed an issue where using invalid parameters in mlxtool perfstat command leads to an infinite waiting time.
	Keywords: mlxtool, perfstat
	Discovered in Release: 5.45
	Fixed in Release: 5.50
1117215	Description: Fixed an issue where the Get-MlnxPCIDeviceSriovSetting command failed on a server with more than one device, when one of the devices was disabled. Following the fix, the command returns results only for the devices that are up.
	Keywords: CIM, SR-IOV
	Discovered in Release: 5.40
	Fixed in Release: 5.50
1118060	Description: Fixed an issue that caused an excessively long installation time when installing the driver on Windows Server 2012 virtual machine in SR-IOV mode with more than 2 VFs.
	Keywords: Installation, virtual machine, VM

Internal Ref.	Issue
	Discovered in Release: 5.40
	Fixed in Release: 5.50
1131167	Description: Fixed an issue that could cause a wrong link state when the PF physical port was 2.
	Keywords: SR-IOV, VF
	Discovered in Release: 5.40
	Fixed in Release: 5.50
1050738	Description: Fixed the issue of when running mlxtool show perfstats, incorrect Rx counters were returned when working in polling mode.
	Keywords: perfstats, mlxtool
	Discovered in Release: 5.40
	Fixed in Release: 5.50
1150078	Description: Fixed a memory leakage that occurred upon requests of 16 million QPs.
	Keywords: Memory leakage, configuration, resources
	Discovered in Release: 5.40
	Fixed in Release: 5.50
989781	Description: Fixed an issue that could cause a black screen on a driver startup in a VM with two VFs or more.
	Keywords: VM, VF, black screen, SR-IOV
	Discovered in Release: 5.40
	Fixed in Release: 5.50
1297549	Description: Fixed a BSOD that occurred while installing WinOF v5.35 due to stack usage overrun.
	Keywords: BSOD, stack usage overrun

Internal Ref.	Issue
	Discovered in Release: 5.35
	Fixed in Release: 5.40.54000
909274/ 896999	Description: RoCE fallback mode is not working when SR-IOV is enabled.
	Keywords: RoCE, fallback mode, SR-IOV
	Discovered in Release: 5.30
	Fixed in Release: 5.40.54000
964757	Description: On servers where Hyper-v role is installed and SR-IOV is enabled, killing the ibdump process will cause a resource leak.
	Keywords: ibdump, resource leak
	Discovered in Release: 5.35
	Fixed in Release: 5.40.54000
1081160	Description: Fixed an issue that caused commands failure and protection domain violation when running the ND application.
	Keywords: ND application, commands failure, protection domain violation
	Discovered in Release: Exists from day one
	Fixed in Release: 5.40.54000
961699	Description: On hypervisor, when one Ethernet port is bind to vmswitch in which SR-IOV is enabled, Network Direct applications do not work on the second port.
	Keywords: hypervisor, vmswitch, SR-IOV
	Discovered in Release: 5.35
	Fixed in Release: 5.40.54000
1064443	Description: Fixed an issue that could cause a system crash on driver load in rare cases. This could occur when the driver was waiting for firmware boot to be completed before accessing the firmware.

Internal Ref.	Issue
	Keywords: system crash, driver load, firmware boot
	Discovered in Release: 5.30
	Fixed in Release: 5.40 54000
1059536	Description: Fixed an issue that allowed executing the QP state change command when the QP was not in a valid state. This caused event viewer flooding.
	Keywords: QP state change, event viewer
	Discovered in Release: 5.40
	Fixed in Release: 5.40.54000
1064443	Description: Fixed an issue that could cause a system crash on driver load in rare cases. This could occur when the driver was waiting for firmware boot to be completed before accessing the firmware.
	Keywords: system crash, driver load, firmware boot
	Discovered in Release: 5.30
	Fixed in Release: 5.40.54000
946432	Description: Fixed an issue that turned the vSwitch port to internal when a bus driver restart was followed by a miniport driver restart.
	Keywords: vSwitch, IPoIB
	Discovered in Release: 5.40
	Fixed in Release: 5.40.54000
1020562	Description: Added an event log when a lost EQE interrupt is detected (event ID 156).
	Keywords: Event log, lost EQE interrupt
	Discovered in Release: 5.40
	Fixed in Release: 5.40.54000
1007026	Description: Fixed an issue where new ND connections cannot be established while ibdump tool is running.

Internal Ref.	Issue
	Keywords: IBDump, ND, RDMA
	Discovered in Release: 5.30
	Fixed in Release: 5.40.54000
1038291	Description: Fixed an issue that caused the ibstat tool to report the wrong link speed. The issue occurred on Ethernet ports when the link speed on the switch was set to 1G and RoCE was enabled.
	Keywords: ibstat, Link Speed, 1G
	Discovered in Release: 5.35
	Fixed in Release: 5.40.54000
987803	Description: Fixed an issue that caused a failure in opening new ND/NDK connections.
	Keywords: ND, NDK, RDMA
	Discovered in Release: 5.30
	Fixed in Release: 5.40.54000
882140	Description: Fixed the IPoIB network interface to show the link's signaling rate.
	Keywords: IPoIB, signaling rate
	Discovered in Release: 5.02
	Fixed in Release: 5.40.54000
866178	Description: Fixed an issue where disabling the driver could cause a hang when opensm.exe was running on the machine.
	Keywords: MAD, IBAL, IBBUS
	Discovered in Release: 5.30
	Fixed in Release: 5.40.54000
964590	Description: Fixed an issue where the VF can remain in an unclear state in case of reset during the loading phase.

Internal Ref.	Issue
	Keywords: VF, SR-IOV
	Discovered in Release: 5.40
	Fixed in Release: 5.40.54000
987804	Description: Fixed an issue where RDMA applications could hang following a miniport restart.
	Keywords: RDMA, miniport restart
	Discovered in Release: 5.30
	Fixed in Release: 5.40.54000
800647	Description: Fixed an issue where during a high CPU load the driver could mistakenly detect a device hang, and issue a NIC reset.
	Keywords: High CPU load, device hang, NIC reset
	Discovered in Release: 5.20
	Fixed in Release: 5.40.54000
584926	Description: Fixed a memory leak in the mlx4u and mlx4nd dll files.
	Keywords: ND, mlx4u, mlx4nd, memory leak
	Discovered in Release: 5.00
	Fixed in Release: 5.40.54000
676863	Description: Fixed an issue that could lead to a high CPU utilization. Following the fix, packets with unrecognized Ethernet protocol are dropped and an appropriate message is printed to the event log.
	Keywords: IPoIB, high CPU utilization
	Discovered in Release: 4.90
	Fixed in Release: 5.40.54000

Internal Ref.	Issue
1005508	Description: Fixed an issue where an ND call to the CancelOverlappedRequests() (Interface IND2Overlapped) function returned an incorrect return value. The fix correctly returns ND_SUCCESS instead of the incorrect ND_PENDING in case of a successful function call.
	Keywords: ND application
	Discovered in Release: 5.40
	Fixed in Release: 5.40.54000
1005018	Description: Fixed an issue that caused ND application crash when creating a shared receive queue, and overloading the queue with post receives.
	Keywords: ND Application, SRQ
	Discovered in Release: 5.30
	Fixed in Release: 5.40.54000
967654	Description: Fixed an issue that could lead to BSOD when removing a Pkey.
	Keywords: Blue screen, IPoIB, Pkey
	Discovered in Release: 5.20
	Fixed in Release: 5.40.54000
1022250	Description: Fixed an issue that could cause BSOD when resizing a number of Queue Pairs (QPs) in Virtual Multiple Machine Queue (VMMQ) mode, while running stress traffic to the VM.
	Keywords: Blue screen, VMMQ
	Discovered in Release: 5.35
	Fixed in Release: 5.40.54000
1022033	Description: Fixed an issue where the internal tracing mechanism could cause driver memory corruption during high stress of commands to the firmware, while writing debug information to the file.
	Keywords: Internal tracing, driver memory corruption, debug information

Internal Ref.	Issue
	Discovered in Release: 5.40
	Fixed in Release: 5.40.54000
1029795	Description: Fixed an issue that could cause a memory leak in the bus driver following multiple resets.
	Keywords: Mlx4_Bus, memory leak
	Discovered in Release: 5.30
	Fixed in Release: 5.40.54000
884771	Description: Fixed an issue where wrong driver hang detection could lead to a miniports reset.
	Keywords: driver hang, mini-ports, reset
	Discovered in Release: 5.30
	Fixed in Release: 5.40.54000
974818	Description: Fixed an issue that wrongly showed RoCE v1 instead of No RoCE as a transport mode in the virtual machine VSTAT output.
	Keywords: VM, guest
	Discovered in Release: 5.40
	Fixed in Release: 5.40.54000
933278/ 935344	Description: Removed the following redundant VM Event Log messages: 122, 104, 144.
	Keywords: VM, Guest
	Discovered in Release: 5.35
	Fixed in Release: 5.40.54000
1000565	Description: Fixed an issue that could cause kernel memory leakage in the Ethernet driver.
	Keywords: Ethernet

Internal Ref.	Issue
	Discovered in Release: 5.02
	Fixed in Release: 5.40.54000
940765	Description: Fixed a wrong Link State value in the VSTAT.
	Keywords: VSTAT
	Discovered in Release: 5.35
	Fixed in Release: 5.40.54000
572934	Description: Fixed an issue where the "mlxtool dbg mstdump" command wrongly returned success value, in case the MST dump feature was disabled.
	Keywords: mlxtool , mstdump
	Discovered in Release: 5.10
	Fixed in Release: 5.40.54000
980191	Description: Fixed an issue that could cause a system crash in a shutdown scenario.
	Keywords: NDIS, system crash, shutdown
	Discovered in Release: 5.40
	Fixed in Release: 5.40.54000
995925	Description: Fixed an issue that occurred rarely when working with LSO - a fragmented packet (with more than 41 fragments) could lead to duplicated headers.
	Keywords: LSO, fragmented packets, duplicated headers
	Discovered in Release: 5.30
	Fixed in Release: 5.35.52000 (FUR 2)
991194	Description: Fixed an issue that caused low bandwidth when using Software vRSS.
	Keywords: Performance, low bandwidth, Software vRSS

Internal Ref.	Issue
	Discovered in Release: 5.30
	Fixed in Release: 5.35.52000 (FUR 2)
966761	Description: Fixed an issue that led to non-optimal Out of box performance for virtual function.
	Keywords: Performance, OOB, SRIOV, virtual function
	Discovered in Release: 5.30
	Fixed in Release: 5.35.12970 (FUR 1)
964639	Description: Fixed an issue which caused a firmware upgrade failure (error code 9) during installation, when RoCE was disabled.
	Keywords: RoCE, firmware upgrade, installation
	Discovered in Release: 5.30
	Fixed in Release: 5.35.12970 (FUR 1)
957390	Description: Fixed an issue where Miniport reset could lead to a driver hang when occurred during driver disabling, or to a system crash when occurred during driver shutdown.
	Keywords: Miniport reset, driver disabling, shutdown
	Discovered in Release: 5.30
	Fixed in Release: 5.35.12965
954467	Description: Fixed an issue where the link speed of the IPoIB adapter was the actual speed and not the official speed (i.e. 54.3GB/s instead of 56 GB/s).
	Keywords: IPoIB adapter, link speed
	Discovered in Release: 5.25
	Fixed in Release: 5.35.12965
936607	Description: Fixed an issue where firmware burning failed on servers with Connectx-3 and Connectx-4 devices.
	Keywords: firmware burning, Connectx-3, Connectx-4

Internal Ref.	Issue
	Discovered in Release: 5.22
	Fixed in Release: 5.35.12965
943258	Description: Fixed an issue where Mellanox counters in Perfmon did not work over HP devices.
	Keywords: Mellanox counters, Perfmon, HP devices
	Discovered in Release: 5.22
	Fixed in Release: 5.35.12965
935523	Description: Fixed an issue where link load of ports connected to virtual machines took more than 10 seconds. the issue occurred on a hyper-v VMQ setup with several virtual machines, and after running massive traffic on the virtual machines.
	Keywords: link load, virtual machines, hyper-v VMQ
	Discovered in Release: 5.30
	Fixed in Release: 5.35.12965
940166	Description: Fixed an issue where in a HyperV machine with VMs running, the network interface required a restart after returning from a Low Power State (sleep/hibernate).
	Keywords: Hyper-v, VMQ, port restart duration
	Discovered in Release: 5.30
	Fixed in Release: 5.35.12965
892647	Description: Fixed an issue that caused the installation process to hang while checking if the RDSH service is installed.
	Keywords: installation, hang, RDSH
	Discovered in Release: 5.22
	Fixed in Release: 5.35.12965
936813	Description: Fixed a driver crash that occurred when the VPORT-ID on the TX packet was invalid.

Internal Ref.	Issue
	Keywords: crash, VPORT-ID, TX packet
	Discovered in Release: 5.30
	Fixed in Release: 5.35.12965
931155	Description: Updated Link Speed reporting when working with IPoIB and booting using WinPE. This issue caused the link to initialize with a 1Gb/s speed instead of the actual speed.
	Keywords: Link Speed, IPoIB, WinPE
	Discovered in Release: 5.30
	Fixed in Release: 5.35.12965
931589	Description: Fixed a rare error that caused a freeze in the error flow during the driver's startup.
	Keywords: mlx4_bus, freeze, startup
	Discovered in Release: 5.30
	Fixed in Release: 5.35.12965
929057	Description: Fixed an SR-IOV team failure caused by an unsuccessful adapter parameters update.
	Keywords: adapter parameters, SR-IOV, teaming
	Discovered in Release: 5.20
	Fixed in Release: 5.35.12965
928030	Description: Fixed an issue which caused a crash in the driver properties dialog, in case more than 8 teaming ports were defined.
	Keywords: crash, driver properties dialog, teaming ports
	Discovered in Release: 5.20
	Fixed in Release: 5.35.12965

2 Introduction

This User Manual describes the installation, configuration and operation of Mellanox WinOF driver.

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 offload (i.e., TCP Segmentation Offload)
- 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 capabilities:

- 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 R2 and above guests
- Virtual Machine Multiple Queue (VMMQ)
- Network Direct Kernel Provider Interface
- PacketDirect Provider Interface

InfiniBand only capabilities:

- SR-IOV over KVM Hypervisor
- Diagnostic tools

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

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.

See also [Document Conventions and Related Documents](#).

2.1 Supplied Packages

Mellanox WinOF driver 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.

2.2 WinOF Set of Documentation

Under <installation_directory>\Documentation:

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

2.3 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: Windows MPI \(MS-MPI\)](#).

3 Installation

3.1 Hardware and Software Requirements

The following are the hardware and software requirements of WinOF driver:

- Windows Operating Systems below:

Description	Package
Windows Server 2012	MLNX_VPI_WinOF-<version>_All_win2012_x64.exe
Windows Server 2012 R2	MLNX_VPI_WinOF-<version>_All_win2012R2_x64.exe
Windows Server 2016	MLNX_VPI_WinOF-<version>_All_win2016_x64.exe
Windows 8.1 Client (64 bit only)	MLNX_VPI_WinOF-<version>_All_win2012R2_x64.exe
Windows 10 Client (64 bit only)	MLNX_VPI_WinOF-<version>_All_win2016_x64.exe

- Administrator privileges on your machine(s)
- Disk space: 100MB

3.2 Downloading Mellanox WinOF Driver

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

- Obtain the machine architecture.

For Windows Server 2012 / 2012 R2 / 2016

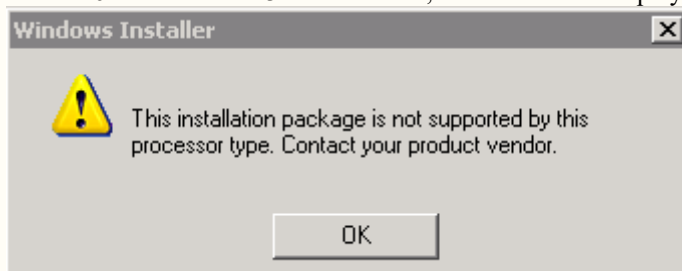
- To go to the Start menu, position your mouse in the bottom-right corner of the Remote Desktop of your screen.
- Open a CMD console (Click Task Manager → File → Run new task, and enter CMD).
- Enter the following command:

```
echo %PROCESSOR_ARCHITECTURE%
```

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

- Go to the Mellanox WinOF web page at <http://www.mellanox.com> → Products → InfiniBand/VPI Drivers → Windows SW/ Drivers.
- 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:



3.3 Installing Mellanox WinOF Driver

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.

⚠ 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 the *WinOF-2 User Manual*.

3.3.1 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-<version>_All_win2016_x64.exe /v"/l*v [LogFile]"
```

Step 3. [Optional] If you do not want to upgrade your firmware version. (MT_SKIPFWUPGRD default value is False.)

```
MLNX_VPI_WinOF-<version>_All_win2016_x64.exe /v" MT_SKIPFWUPGRD=1"
```

Step 4. [Optional] If you want to control the installation of the WMI/CIM provider. (MT_WMI default value is True.)

```
MLNX_VPI_WinOF-<version>_All_win2016_x64.exe /v" MT_WMI=1"
```

Step 5. [Optional] If you want to control whether to restore network configuration or not. (MT_RESTORECONF default value is True.)

```
MLNX_VPI_WinOF-<version>_All_win2016_x64.exe /v" MT_RESTORECONF=1"
```

For further help, please run:

```
MLNX_VPI_WinOF-<version>_All_win2016_x64.exe /v" /h"
```

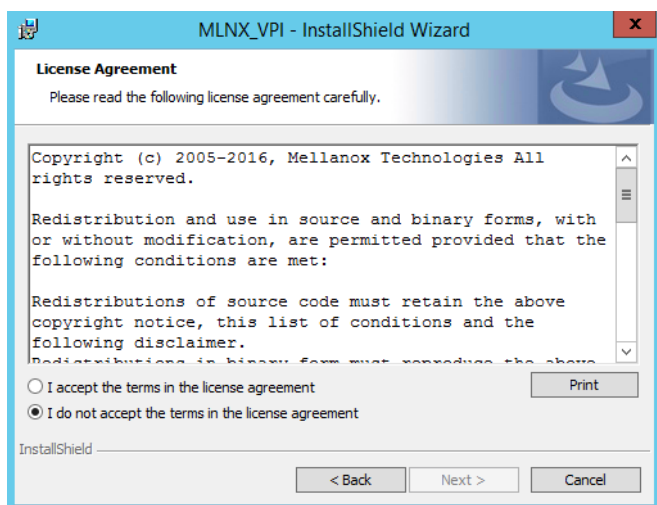
Step 6. [Optional] If you want to control the driver's loading timeout value before burning the firmware, run:

```
MLNX_VPI_WinOF-<version>_All_win2016_x64.exe /v" /MMT_DRIVER_LOAD_TIMEOUT=30"
```

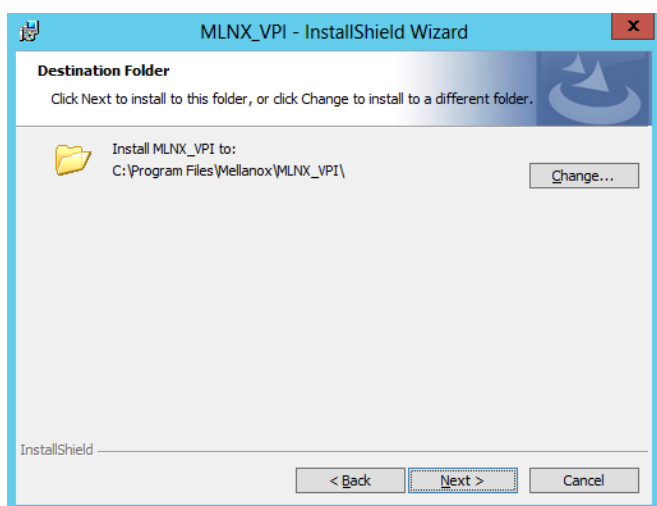
Note: Timeout value is in seconds, and the default value is 120.

Step 7. Click **Next** in the Welcome screen.

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



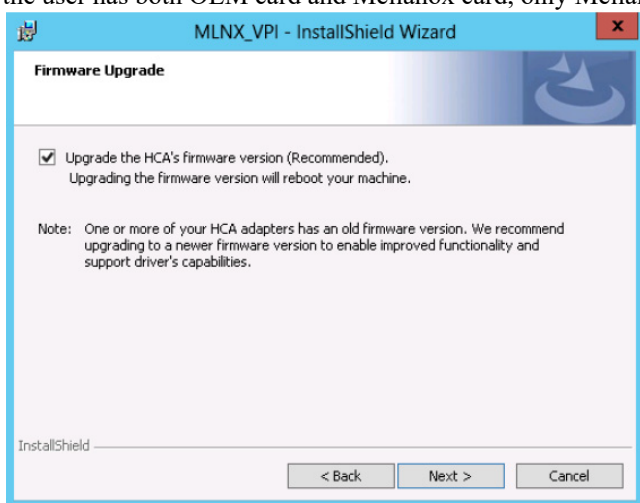
Step 9. Select the target folder for the installation.



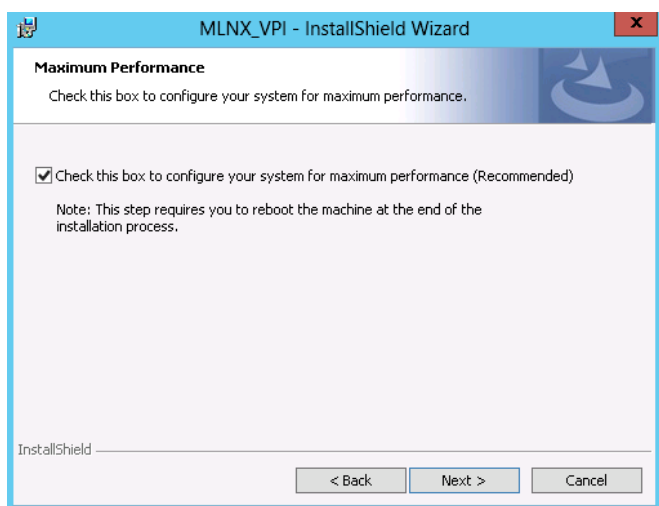
Step 10. The firmware upgrade screen will be displayed in the following cases:


- 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.



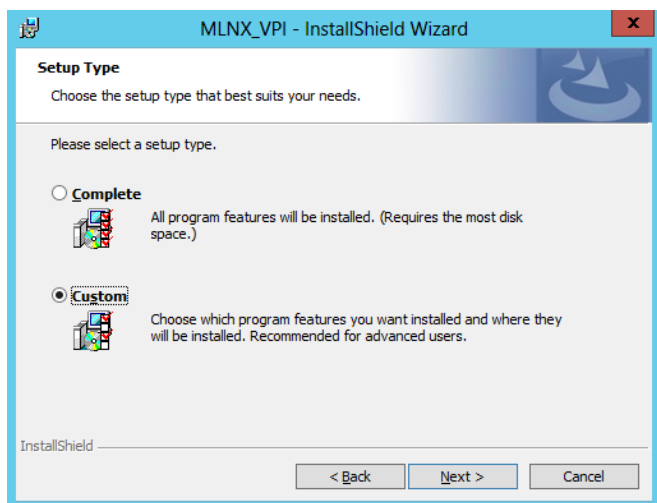
Step 11. Configure your system for maximum performance by checking the maximum performance box.



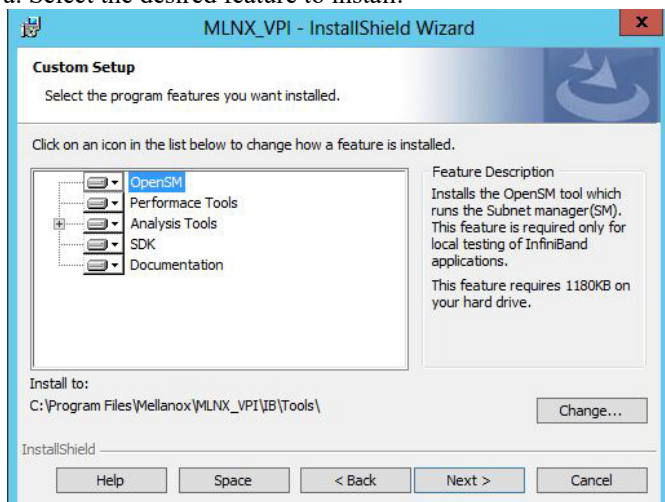
 This step requires rebooting your machine at the end of the installation.

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.

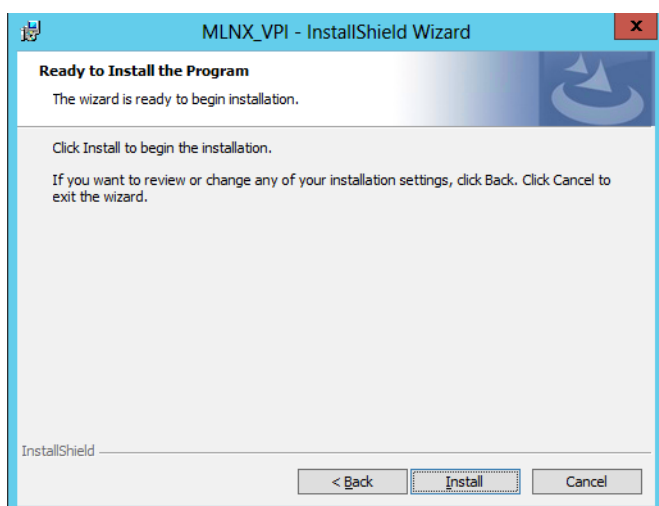


a. Select the desired feature to install:

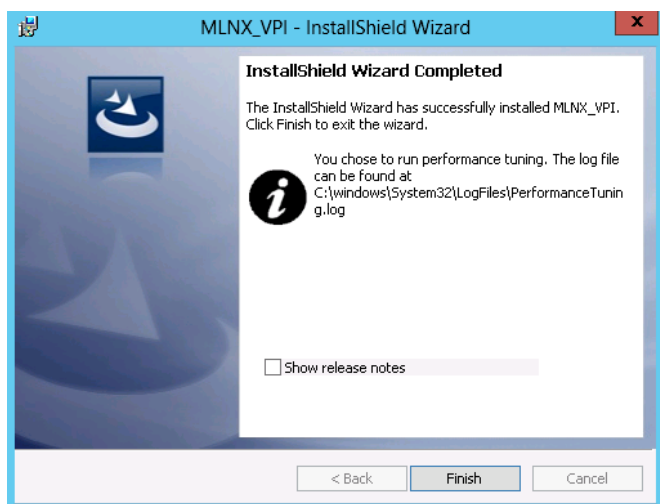


- OpenSM - installs Windows OpenSM that is required to manage the subnet from a host. OpenSM is part of the driver and installed automatically.
- Performance tools - install the performance tools that are used to measure the InfiniBand performance in user environment.
- Analyze tools - install the tools 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.

b. Click **Install** to start the installation.



Step 13. Click **Finish** to complete the installation.



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



3.3.2 Unattended Installation

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

- ⚠ 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.

Step 1. Open a CMD console. [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-<version>_All_win2016_x64.exe /S /v/qn
```

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

```
> MLNX_VPI_WinOF-<version>_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. (MT_SKIPFWUPGRD default value is False.)

```
MLNX_VPI_WinOF-<version>_All_win2016_x64.exe /v" MT_SKIPFWUPGRD=1"
```

Step 5. [Optional] If you wish to control the installation of the WMI/CIM provider. (MT_WMI default value is True.)

```
> MLNX_VPI_WinOF-<version>_All_win2016_x64.exe /v" /MT_WMI=1"
```

Step 6. [Optional] If you wish to control whether to restore network configuration or not. (MT_RESTORECONF default value is True.)

```
> MLNX_VPI_WinOF-<version>_All_win2016_x64.exe /v" MT_RESTORECONF=1"
```

For further help, please run:

```
> MLNX_VPI_WinOF-<version>_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-<version>_All_win2016_x64.exe /v" /MMT_DRIVER_LOAD_TIMEOUT=30"
```

Note: 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. (PERFCHECK default value is True.)

```
> MLNX_VPI_WinOF-<version>_All_win2016_x64.exe /vPERFCHECK=0 /vPERFCHECK=0
```

Step 9. [Optional] If you wish to control whether to install ND provider or not. (MT_NDPROPERTY default value is True.)

```
MLNX_VPI_WinOF-<version>_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 [Setup Return Codes](#) in the Troubleshooting section.


3.4 Uninstalling Mellanox WinOF Driver

3.4.1 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 [Supplied Packages](#) 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.

3.4.2 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:

1. Open a CMD console
[Windows Server 2012 / 2012 R2/ 2016]: Click Start → Task Manager → File → Run new task → and enter CMD.
2. Uninstall the driver. Run:

```
> MLNX_VPI_WinOF-<version>_All_win2016_x64.exe /x /v/qn
```

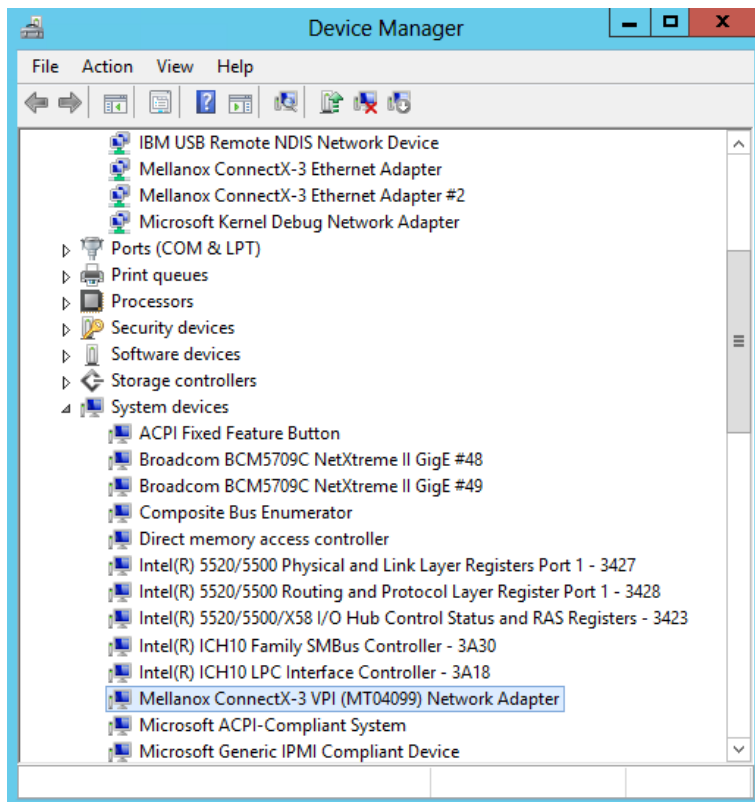
3.5 Installation Results

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”.



3.6 Extracting Files without Running Installation

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

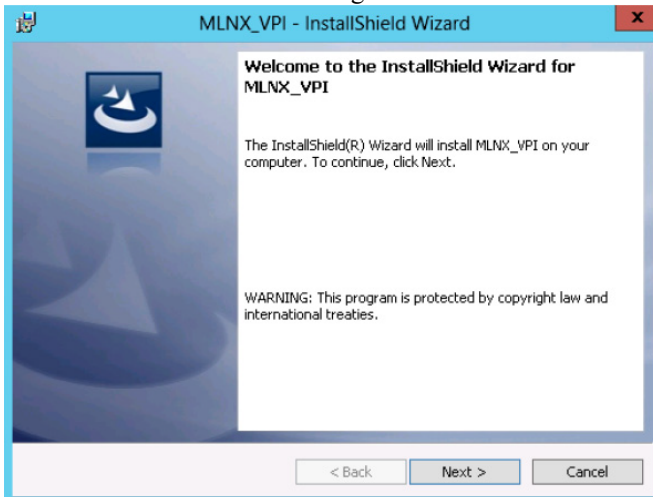
1. Open a CMD console
[Windows Server 2012 / 2012 R2/ 2016]: Click Start → Task Manager → File → Run new task → and enter CMD.
2. Extract the driver and the tools:

```
MLNX_VPI_WinOF-<version>_All_win2016_x64.exe /a
```

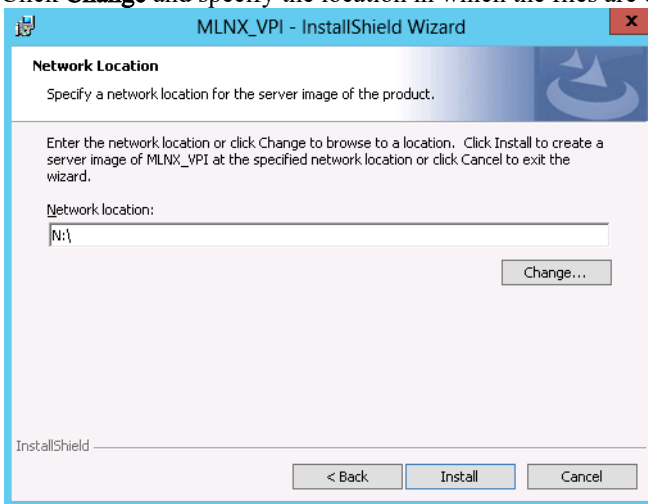
- To extract the driver files only:

```
MLNX_VPI_WinOF-<version>_All_win2016_x64.exe /a /vMT_DRIVERS_ONLY=1
```

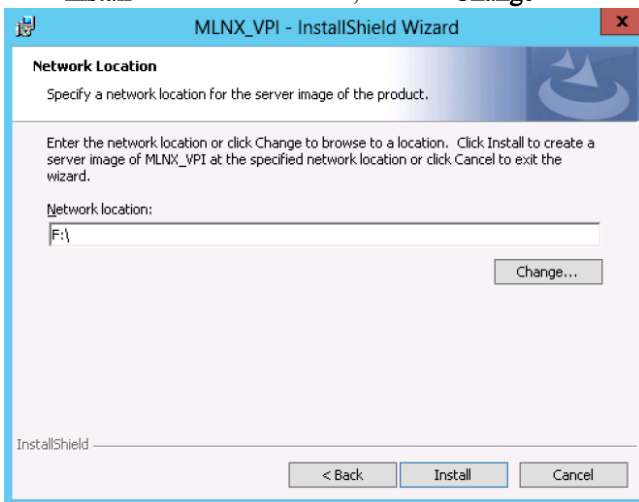
3. Click **Next** to create a server image.



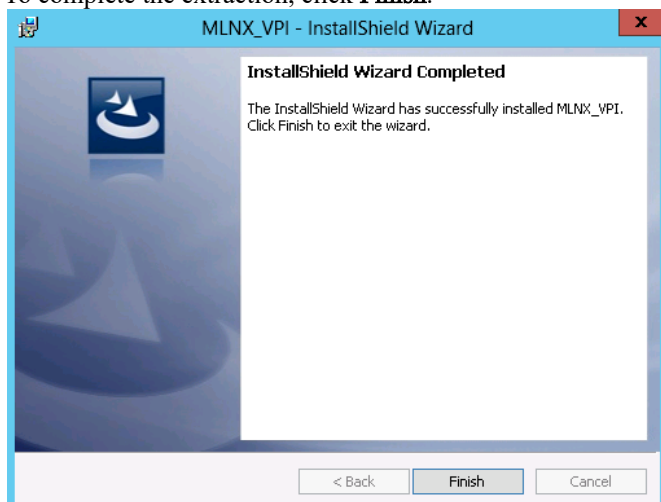
4. Click **Change** and specify the location in which the files are extracted to.



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



6. To complete the extraction, click **Finish**.



3.7 Upgrading Firmware & WinOF Driver

3.7.1 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 the *MFT User Manual* at www.mellanox.com → Products → InfiniBand/VPI Drivers → Firmware Tools.

3.7.2 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 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 [RoCE Default Configuration](#).

3.8 Booting Windows from an iSCSI Target or PXE

3.8.1 Configuring the WDS, DHCP and iSCSI Servers

3.8.1.1 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.

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

Note: Use 'index:2' for Windows setup and 'index:1' for WinPE.

4. Add the Mellanox driver to install.wim.

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

Note: 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'.

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>

3.8.1.2 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.

3.8.1.3 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:

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.

3.8.2 Configuring the Client Machine

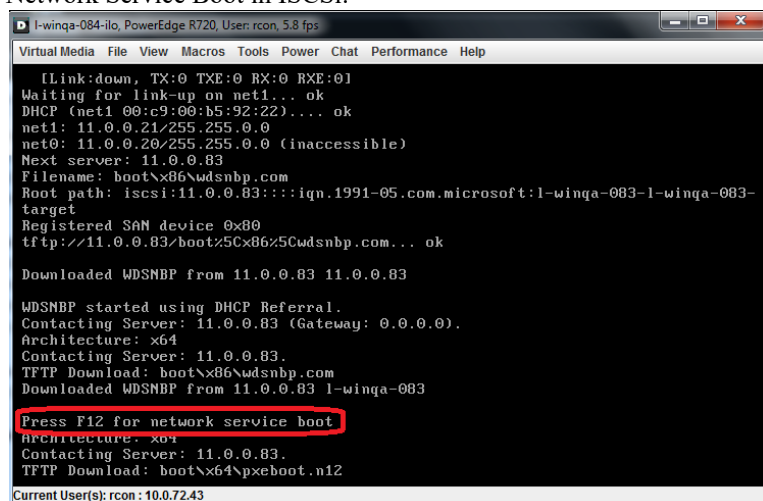
To configuring your client:

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.

3.8.3 Installing the Operating System

1. Reboot your client.
2. Press **F12** when asked to proceed to network boot.

Network Service Boot in iSCSI:



```

I-winqa-084-ilo, PowerEdge R720, User: rcon, 5.8 fps
Virtual Media File View Macros Tools Power Chat Performance Help

[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 SAM device 0x80
tftp://11.0.0.83/boot%5C%86%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

Current User(s): rcon:10.0.72.43
  
```

Network Service Boot in PXE:


```

I-winqa-084-ilo, PowerEdge R720, User: rcon, 5.8 fps
VirtualMedia File View Macros Tools Power Chat Performance Help

[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\x86\wdsnbp.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 I-winqa-083

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

Current User(s): rcon : 10.0.72.43

```

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

```

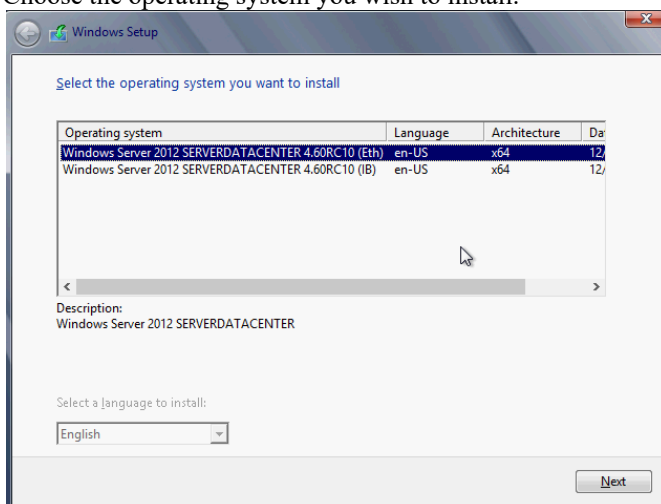
Windows Boot Manager (Server IP: 11.0.0.83)

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)

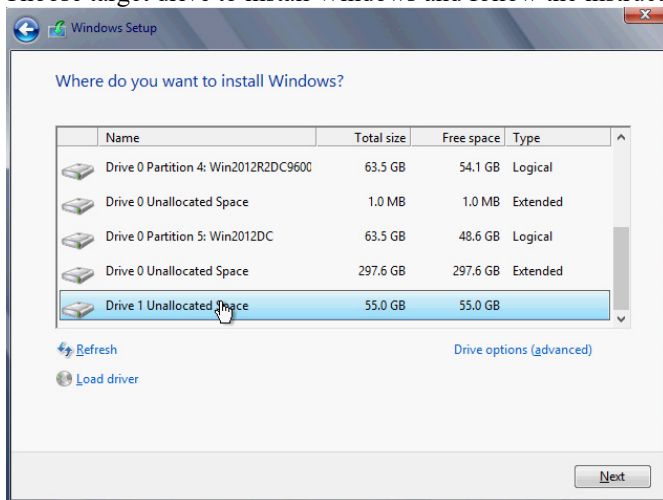
```

- Choose the operating system you wish to install.



- Run the Windows Setup Wizard.

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



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

4 Features Overview and Configuration

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

❗ Changes made to the Windows registry take immediate effect, and no backup is automatically made. Do not edit the Windows registry unless you are confident regarding the changes.

4.1 Ethernet Network

This section describes Ethernet network features and their configuration.

- [Port Configuration](#)
- [Assigning Port IP After Installation](#)
- [56GbE Link Speed](#)
- [RDMA over Converged Ethernet \(RoCE\)](#)
 - [RoCEv2](#)
 - [RoCE v2 UDP Port](#)
 - [Configuring RoCE](#)
- [Teaming and VLAN](#)
- [Command Line Based Teaming Configuration](#)
- [Header Data Split](#)
- [Receive Segment Coalescing \(RSC\)](#)
- [Configuring Quality of Service \(QoS\)](#)
- [Configuring the Ethernet Driver](#)
- [Differentiated Services Code Point \(DSCP\)](#)
- [Lossless TCP](#)
- [Receive Side Scaling \(RSS\)](#)
- [Ignore Frame Check Sequence \(FCS\) Errors](#)
- [VXLAN](#)
- [Threaded DPC](#)

4.1.1 Port Configuration

4.1.1.1 Auto Sensing

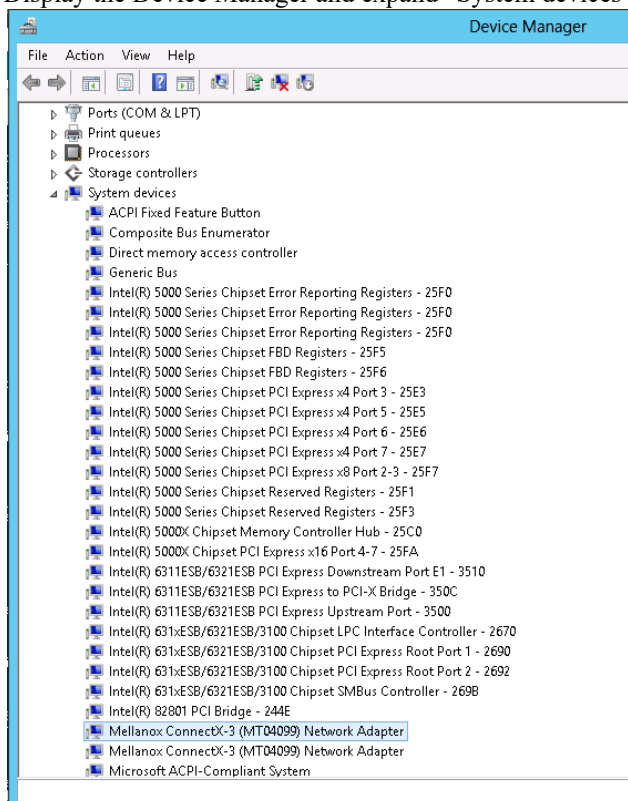
Auto Sensing enables the NIC to automatically sense the link type (InfiniBand or Ethernet) based on the cable connected to the port and load the appropriate driver stack (InfiniBand or Ethernet).

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 the “Port Protocol Configuration” section below.

4.1.1.2 Port Protocol Configuration

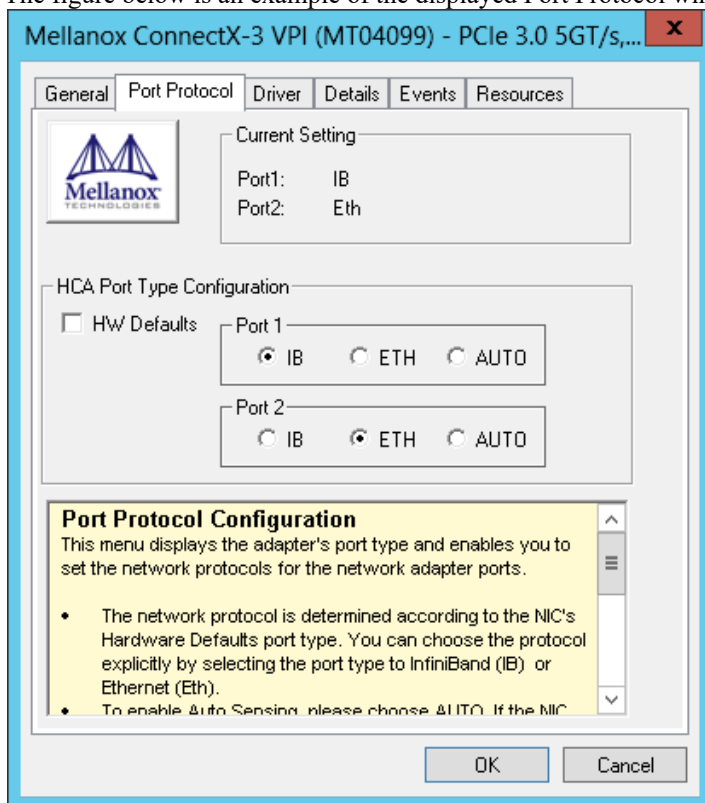
1. Display the Device Manager and expand “System devices”.



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.



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

- Choose HW Defaults option. In this case, the drivers behavior is as follow:
 - If HW Default option is chosen, the driver will use the Auto Sense option by default
 - If the Auto Sense option fails, 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 the adapter card. If this configuration is created occasionally by auto sensing, the driver will fail to startup. If this configuration is intentionally defined as explained in Step 2 above, the driver will set RoCE mode to v1.5 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 check box.
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*.

4.1.2 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:

1. Open a CMD console
[Windows Server 2012 / 2012 R2/ 2016]: Click Start → Task Manager → File → Run new task → and enter CMD.

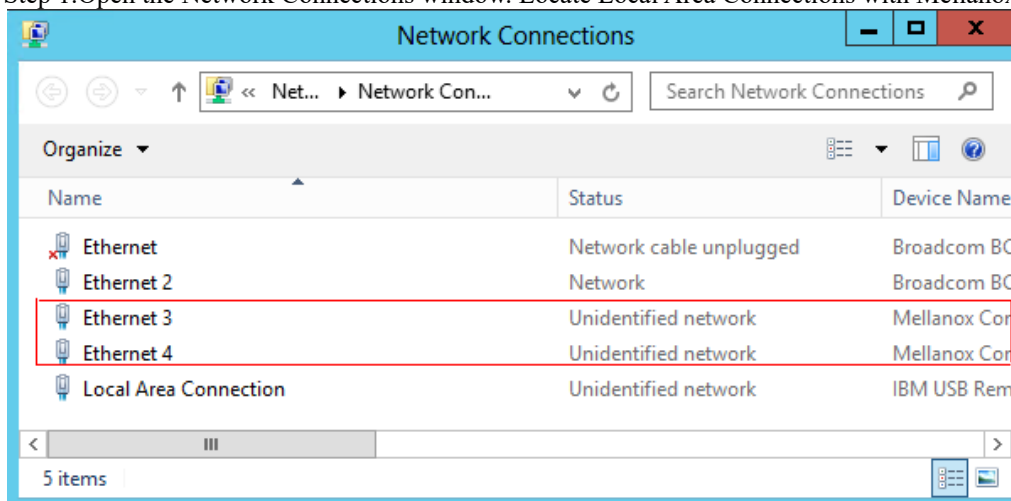
2. Display the MAC address as “Physical Address”

```
> ipconfig /all
```

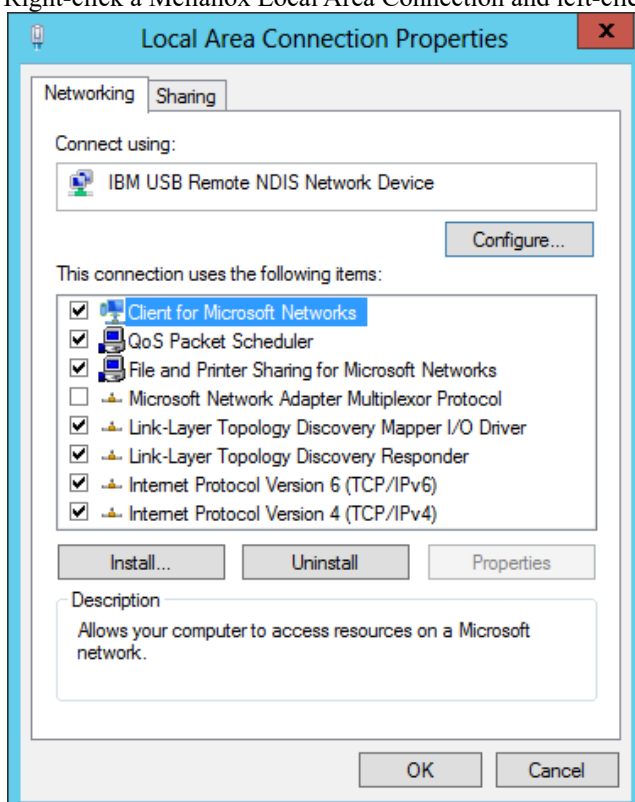
Configuring a static IP is the same for both IPoIB and Ethernet adapters.

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

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

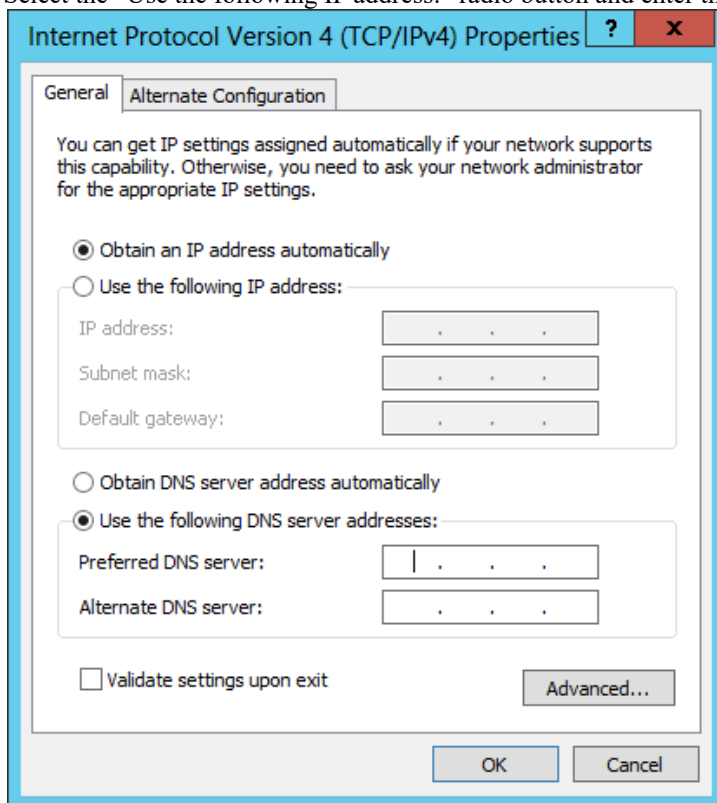


2. Right-click a Mellanox Local Area Connection and left-click Properties.



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

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



5. Click **OK**.
6. Close the Local Area Connection dialog.
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 . . . . . :
...
```

4.1.3 56GbE Link Speed

System Requirements

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

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.

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

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

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) #
```

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) #
```

4.1.4 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.

4.1.4.1 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>

4.1.4.1.1 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 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 see [Port Protocol Configuration](#).

4.1.4.1.2 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 [Configuring Quality of Service \(QoS\)](#).

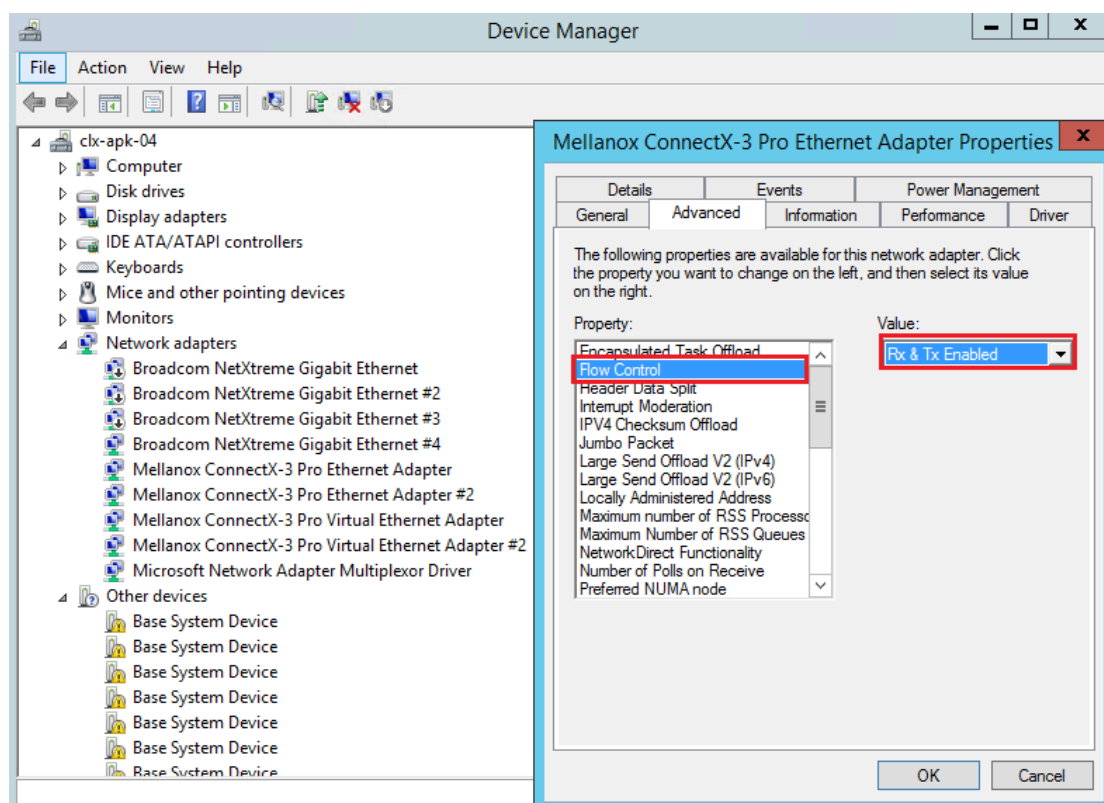
4.1.4.1.2.1 Global Pause (Flow Control)

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.



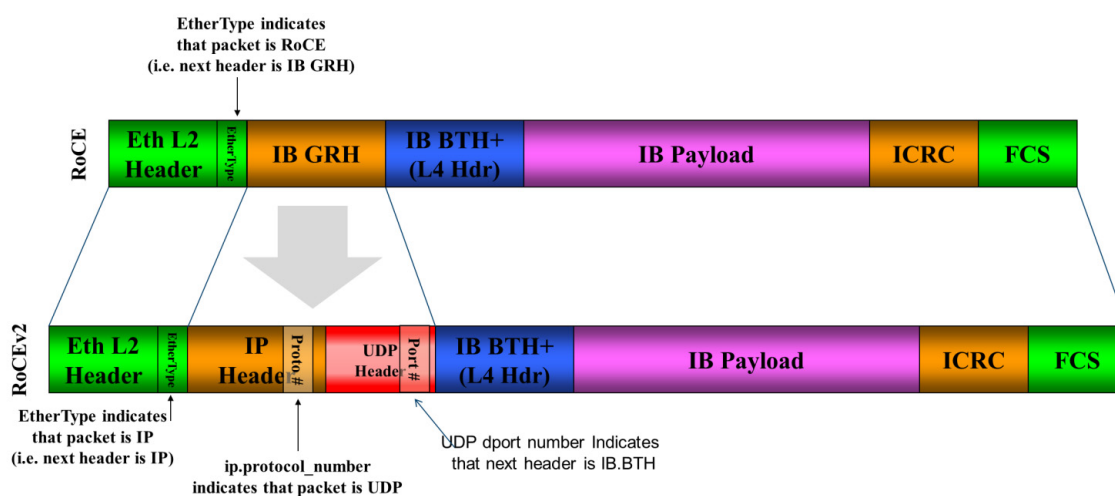
4.1.4.2 RoCEv2

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.

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.

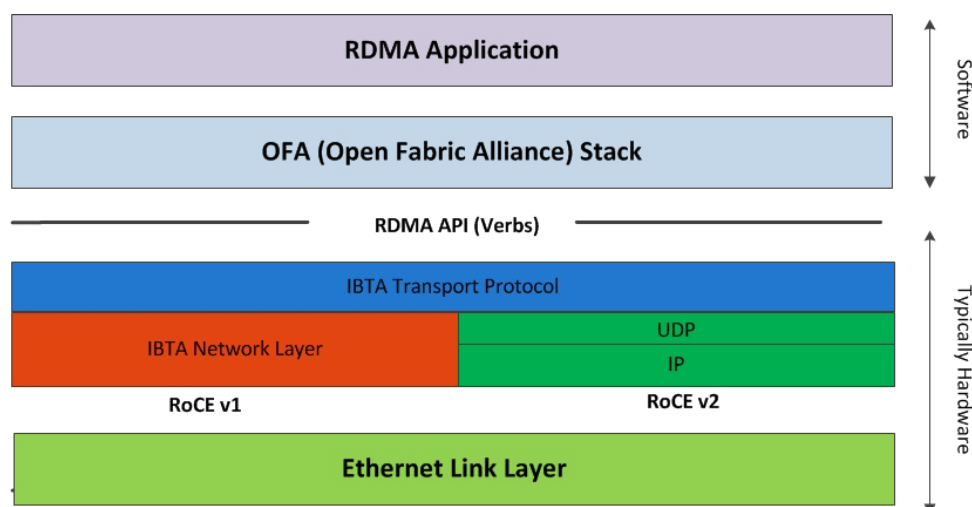
The UDP source port is calculated as follows: $\text{UDP.SrcPort} = (\text{SrcPort} \text{ XOR } \text{DstPort}) \text{ OR } 0xC000$, where SrcPort and DstPort are the ports used to establish the connection.

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 the figure above "RoCE and RoCE v2 Frame Format Differences"), in a completely transparent way⁽¹⁾.

Note (1): Standard RDMA APIs are IP based already for all existing RDMA technologies.

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.

4.1.4.3 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 the [RoCE Options](#) table.

4.1.4.3.1 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.

4.1.4.4 Configuring RoCE

4.1.4.4.1 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 the *SwitchX User Manual*.

4.1.4.4.2 Configuring Arista Switch

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

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

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

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

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

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

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
```

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
```

4.1.4.4.2.1 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
```

4.1.4.4.2.2 Using Priority Flow Control (PFC)

To enable PFC 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
```

4.1.4.4.3 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.


4.1.4.4.3.1 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.

4.1.4.4.4 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
```

4.1.4.4.5 How to Configure Storage Space Direct over RDMA

Storage Spaces Direct uses industry-standard servers with local-attached drives to create highly available, highly scalable software-defined storage at a fraction of the cost of traditional SAN or NAS arrays.

To configure storage space direct over RDMA:

1. Deploy storage space direct. For further information on how to do so, see:
<https://docs.microsoft.com/en-us/windows-server/storage/storage-spaces/deploy-storage-spaces-direct>.
2. Configure the host environment.
 - a. Install DCB module, enable DCB in willing mode and remove all previous QoS configuration.

```
Install-WindowsFeature "data-center-bridging"
Remove-NetQosTrafficClass
Remove-NetQosPolicy -Confirm:$False
Set-NetAdapterQos -Enabled 1 *
Set-NetQosDcbxSetting -Willing 0 -Confirm:$false
```

- b. Set the QoS rules, mapping SMB to priority 3 (RoCE).

```
New-NetQosPolicy "SMB" -NetDirectPortMatchCondition 445 -PriorityValue8021Action3 -
PolicyStore activestore -Confirm:$false
New-NetQosPolicy "SMB" -NetDirectPortMatchCondition 445 -PriorityValue8021Action3 -
Confirm:$false
New-NetQosPolicy "DEFAULT" -PriorityValue8021Action 3 -PolicyStore activestore-
Confirm:$false -Default
New-NetQosPolicy "DEFAULT" -PriorityValue8021Action 3 -Confirm:$false -Default
New-NetQosPolicy "TCP" -IPProtocolMatchCondition TCP -PriorityValue8021Action0 -
DSCPAction 0 -PolicyStore activestore -Confirm:$false
New-NetQosPolicy "TCP" -IPProtocolMatchCondition TCP -PriorityValue8021Action0 -
DSCPAction 0 -Confirm:$false
New-NetQosPolicy "UDP" -IPProtocolMatchCondition UDP -PriorityValue8021Action0 -
PolicyStore activestore -Confirm:$false
New-NetQosPolicy "UDP" -IPProtocolMatchCondition UDP -PriorityValue8021Action0 -
Confirm:$false
```

- c. Enable PFC on priority 3.

```
Disable-NetQosFlowControl 0,1,2,3,4,5,6,7
```

3. Configure RoCE v2 for ConnectX-3 Pro using Mellanox SwitchX switches is needed.
<https://community.mellanox.com/docs/DOC-1444>

For additional information, see: <https://community.mellanox.com/docs/DOC-2283>

4.1.4.4.6 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.

4.1.5 Teaming and VLAN

Windows Server 2012 and above supports Teaming as part of the operating system.

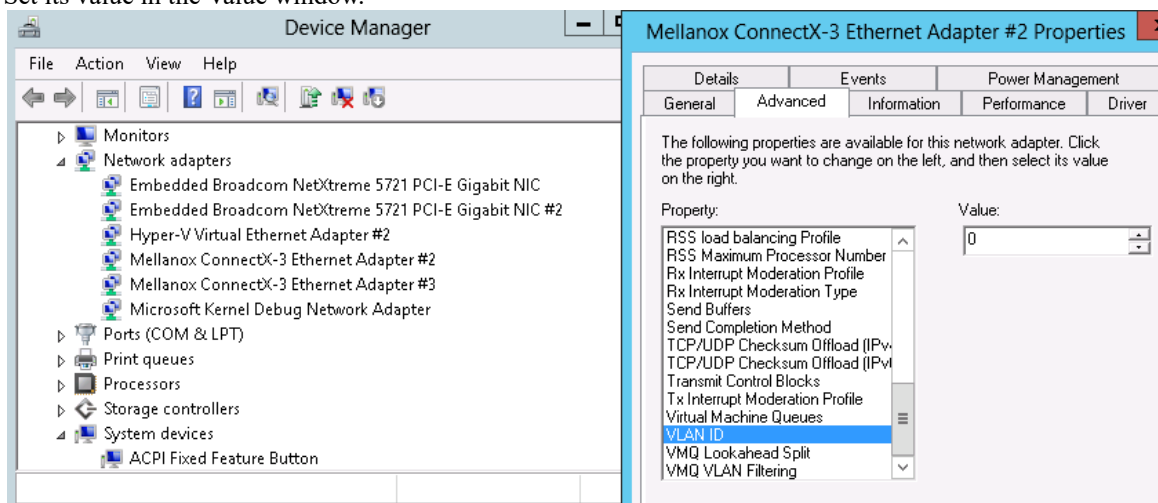
Please refer to Microsoft guide “NIC Teaming in Windows Server 2012” at the following link: https://technet.microsoft.com/en-us/windows-server-docs/networking/technologies/nic-teaming/nic-teaming?f=255&MSPPError=-2147217396#bkmk_over

4.1.5.1 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.

1. Open the Device Manager.
2. Go to the Network adapters.
3. Right click Properties on Mellanox ConnectX®-3 Ethernet Adapter card.
4. Go to Advanced tab.
5. Choose the VLAN ID in the Property window.
6. Set its value in the Value window.



4.1.6 Command Line Based Teaming Configuration

4.1.6.1 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}
```

4.1.6.2 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
```

4.1.6.3 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
```


4.1.6.4 Attach Members to Team

```
C:\Users\Administrator\Desktop>vlan_config attach team MyTeam
{30C354DB-62E9-41CBB709-11063FAF4E95} 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
```

4.1.6.5 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
```

4.1.6.6 Modify a Team

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

4.1.6.7 Modify a VLAN

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

4.1.6.8 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
```

4.1.6.9 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
```

4.1.6.10 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
```

4.1.6.11 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
```

4.1.6.12 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}
```

4.1.6.13 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
```

4.1.7 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.

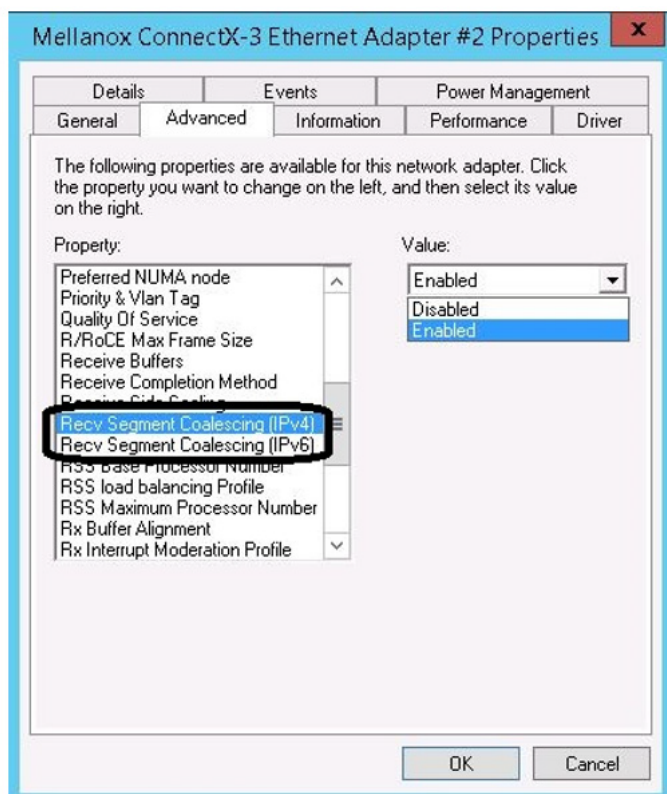
For further information, please refer to the MSDN library:

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

4.1.8 Receive Segment Coalescing (RSC)

Processing packets on the receive side for Ethernet is done on a per-packet basis by the miniport driver and the stack above it. The RSC feature reduces the CPU utilization when the driver coalesces segments belonging to the same TCP connection, and indicated as one big packet. This feature is enabled in the driver by default for both IPv4 and IPV6, and for both Physical Function and Virtual Function.

It is advisable to disable the RSC feature when running latency sensitive traffic.



4.1.8.1 System Requirements

Supported Operating Systems:

- Windows Server 2012
- Windows Server 2012 R2
- Windows Server 2016

The above specified operating systems flavors support the feature by default. They must be enabled in Windows Client 8.0, 8.1 and 10, using netsh commands.

4.1.8.2 RSC Counters

The full RSC counters list is specified in the [Mellanox Adapter Diagnostics Counters](#) table. All counters apply to Ethernet ports only.

RSC Aborts	Number of RSC abort events. That is, the number of exceptions other than the IP datagram length being exceeded. This includes the cases where a packet is not coalesced due to insufficient hardware resources.
RSC Coalesce Events	Number of RSC coalesced events. That is, the total number of packets that were formed from coalescing packets.
RSC Coalesce Octets	Number of RSC coalesced bytes.
RSC Coalesce Packets	Number of RSC coalesced packets.

The RSC counters can also be viewed via PowerShell, as shown in the following example:

```
PS D:\Users\Administrator> (Get-NetAdapterStatistics).RscStatistics

CoalescedBytes       : 17529121153
CoalescedPackets     : 12247142
CoalescingEvents      : 1477183
CoalescingExceptions  : 0
PSComputerName       :
```

4.1.9 Configuring Quality of Service (QoS)

4.1.9.1 System Requirements

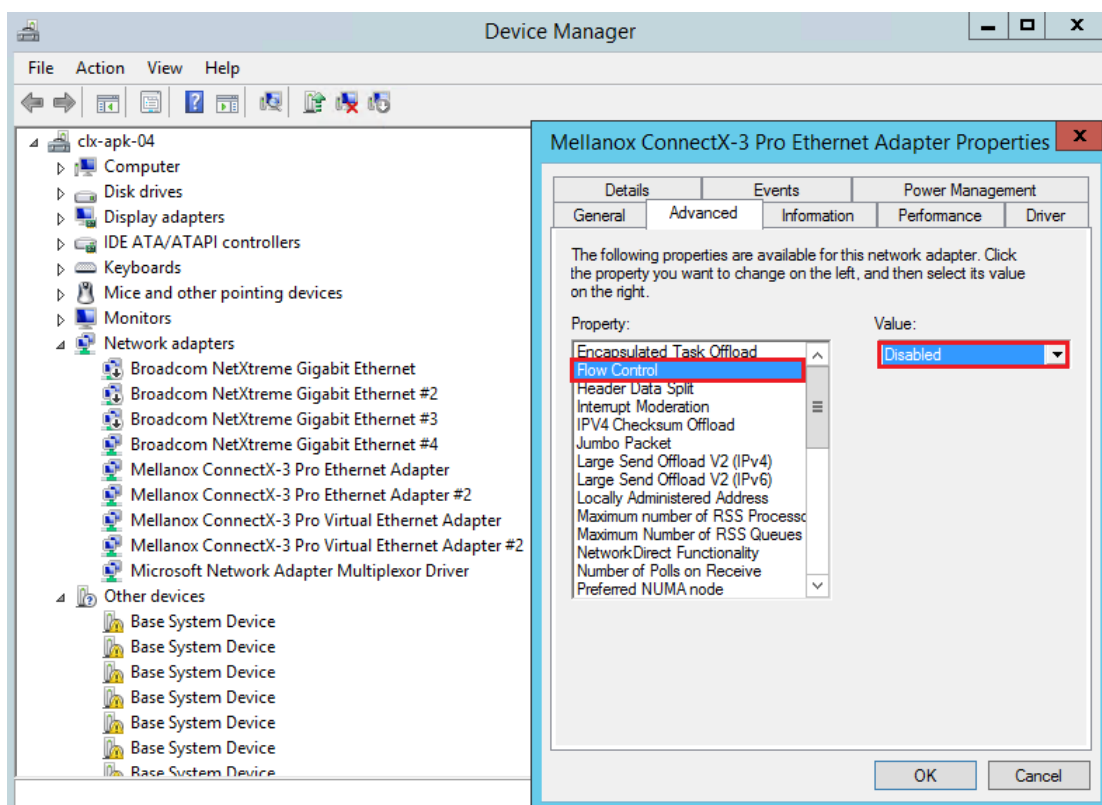
Operating Systems: Windows Server 2012, Windows Server 2012 R2 and Windows Server 2016.

4.1.9.2 QoS Configuration

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

4.1.9.2.1 To Disable Flow Control Configuration

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



4.1.9.2.2 To Install the Data Center Bridging using the Server Manager

1. Open the 'Server Manager'.
2. Select 'Add Roles and Features'.
3. Click Next.
4. Select 'Features' on the left panel.

5. Check the 'Data Center Bridging' checkbox.
6. Click 'Install'.

4.1.9.2.3 To Install the Data Center Bridging using PowerShell

Enable Data Center Bridging (DCB).

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

4.1.9.2.4 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.

1. Change the Windows PowerShell execution policy. To change the execution policy, please refer to Step 1 in [PowerShell Configuration](#).
2. Remove the entire previous QoS configuration.

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

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

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

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
```

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
```

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" -
RegistryValue "55"
```

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 -Confirm:$false
PS $ New-NetIPAddress -InterfaceAlias "Ethernet 4" -IPAddress 192.168.1.10 -Prefix-Length 24 -Type Unicast
```

8. 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.

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

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

10. Enable QoS on the relevant interface.

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

11. Enable PFC on priority 3.

```
PS $ Enable-NetQosFlowControl -Priority 3
PS $ New-NetQosPolicy -name "SMB class" -Priority 3
```

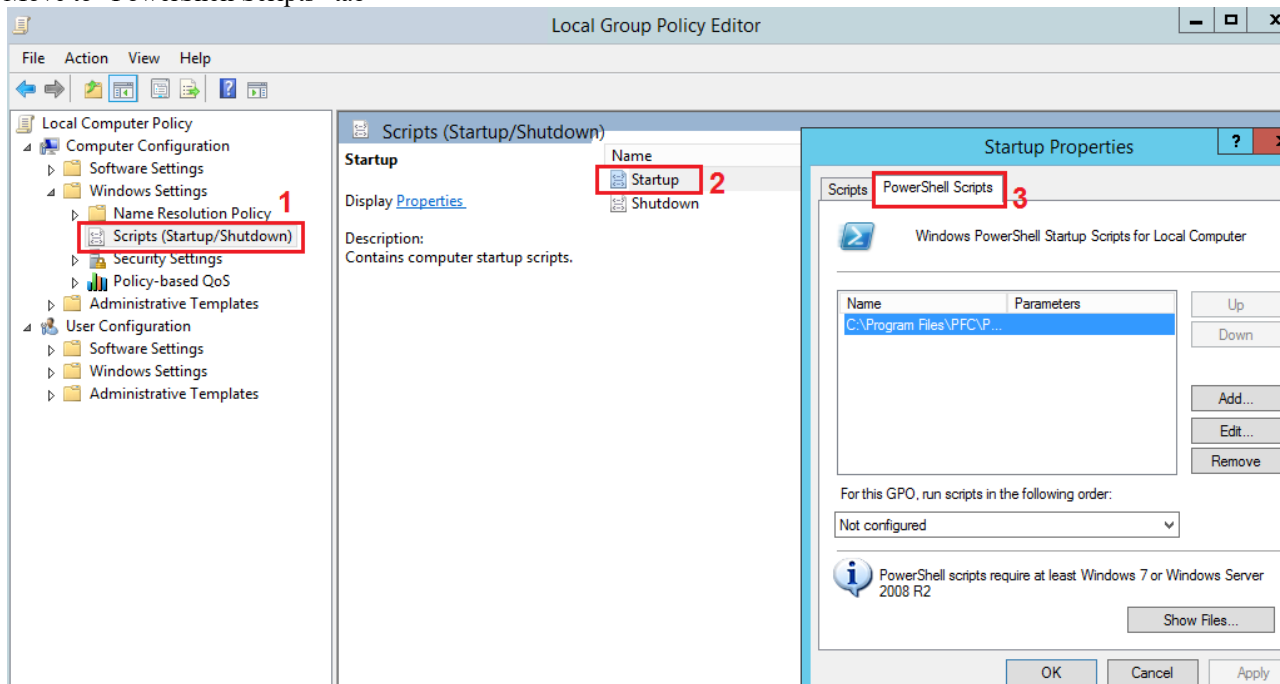
12. Configure Priority 3 to use ETS.

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

4.1.9.2.5 To Add the Script to the Local Machine Startup Scripts

1. From the PowerShell invoke.
2. In the pop-up window, under the 'Computer Configuration' section, perform the following:
 - a. Select Windows Settings
 - b. Select Scripts (Startup/Shutdown)
 - c. Double click Startup to open the Startup Properties

- d. Move to “PowerShell Scripts” tab



- e. 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 -
PriorityValue8021Action 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
```

- f. Browse for the script's location.
g. Click OK
h. To confirm the settings applied after boot run:

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

4.1.9.3 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>

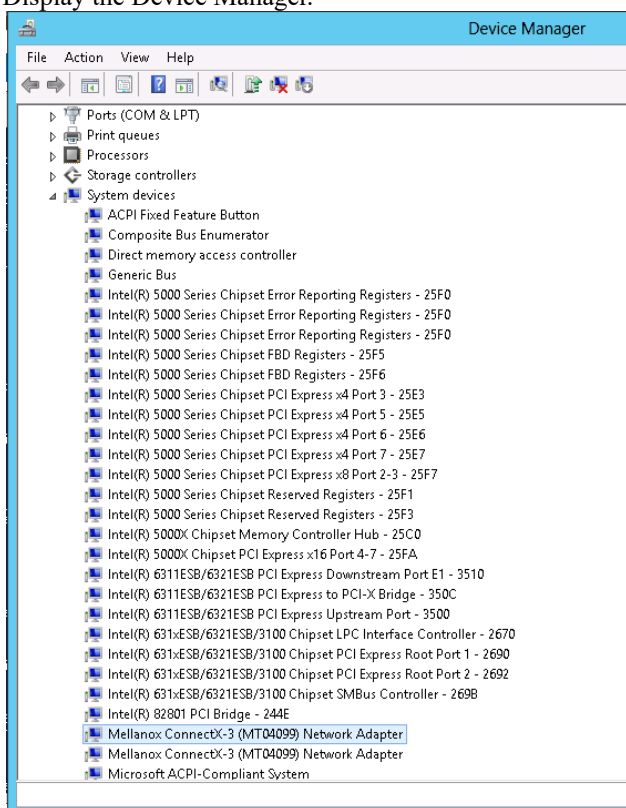
For further details on configuring ETS on Windows™ Server, please refer to:

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

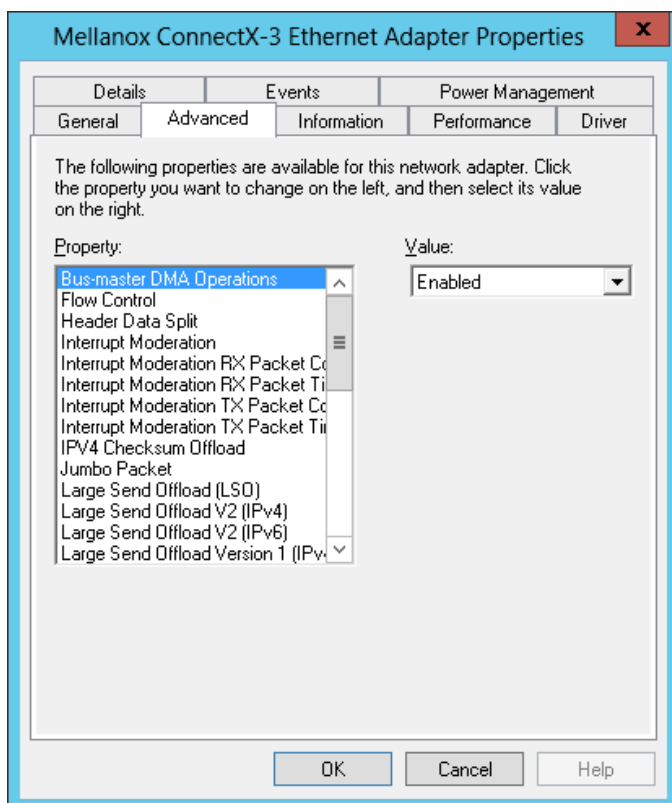
4.1.10 Configuring the Ethernet Driver

The following steps describe how to configure advanced features.

1. Display the Device Manager.



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



3. Modify configuration parameters to suit your system.

Please note the following:

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

4.1.11 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.

4.1.11.1 System Requirements

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

4.1.11.2 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

4.1.11.3 Configuring Quality of Service for TCP and RDMA Traffic

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

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

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

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

3. Configure DCB.

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

4. Enable Network Adapter QoS.

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

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

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

4.1.11.4 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
```

4.1.11.5 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 -PolicyStore
activestore
```

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

```
PS $ New-NetQosPolicy "TCP1" -DSCPAction 16 -IPDstPortStartMatchCondition 31000 -IPDstPortEndMatchCondition 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 -IPDstPortEndMatchCondition 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 -AlgorithmETS
PS $ New-NetQosTrafficClass -name "TCP2" -priority 5 -bandwidthPercentage 80 -AlgorithmETS
```

4.1.11.6 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 - PriorityValue8021Action 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

4.1.11.7 Registry Settings

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

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.

Registry Key	Description
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.

4.1.11.7.1 Default Settings

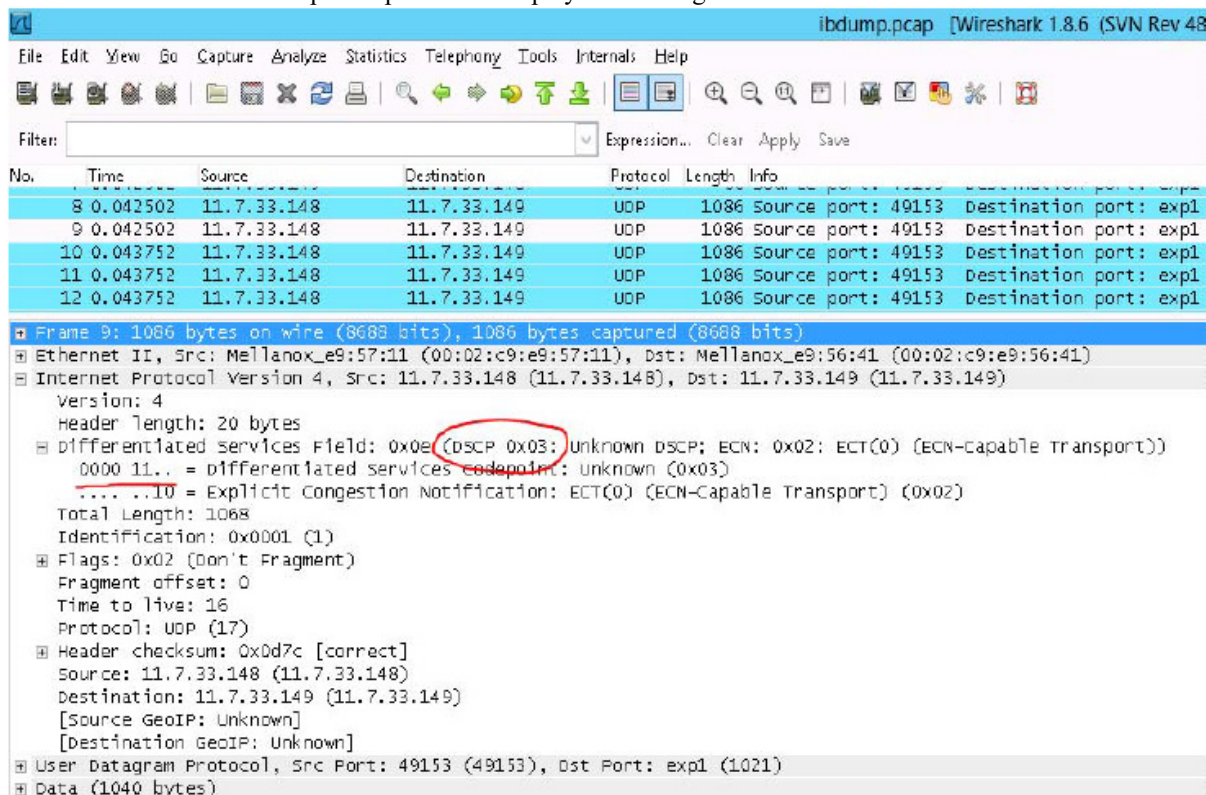
When DSCP configuration registry keys are missing in the miniport registry, the following defaults are assigned:

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

4.1.11.8 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.



The image shows a Wireshark packet capture of traffic from 11.7.33.148 to 11.7.33.149. The selected packet (Frame 9) is a UDP packet with a length of 1086 bytes. The packet details pane shows the following structure:

- 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: 0x03 (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: 1086
 - 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: exp1 (1021)
- Data (1040 bytes)

4.1.12 Lossless TCP

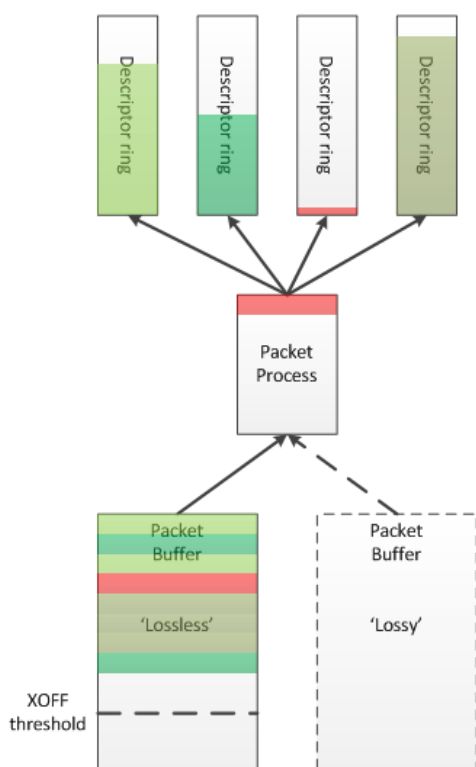
4.1.12.1 System Requirements

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

4.1.12.2 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.

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 – drop mode and poll mode.

4.1.12.2.1 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).

4.1.12.2.2 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.

4.1.12.2.3 Default Behavior

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

4.1.12.3 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.

4.1.12.4 System Requirements

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

4.1.12.5 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.

4.1.12.5.1 Enabling Lossless TCP Using The Registry Key DelayDropTimeout

Registry key location:

```
HKLM\SYSTEM\CurrentControlSet\Control\Class\Class\{4d36e972-e325-11cebfc1-08002be10318}\<nn>\DelayDropTimeout
```

For instructions on how to find interface index in registry <nn>, Please refer to [Finding the Index Value of the Network Interface](#).

Enabling Lossless TCP Using The Registry Key DelayDropTimeout

Key Name	Key Type	Values	Description
DelayDrop Timeout	REG_D WORD	0 = disabled (default) 1-65535 = enabled 0	<p>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 milliseconds (+/- 200 microseconds).</p> <p>Choosing the value of 65535 enables the feature but the amount of time a packet may wait for a free descriptor is infinite.</p> <p>Note: Changing the value of the DelayDropTimeout registry key requires restart of the network interface</p>

4.1.12.5.2 Entering/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

4.1.12.6 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 the table below.

Lossless TCP Associated Events

Event ID	Event Description
0x0057 <Device Name>	Dropleless mode entered on port <X>. Packets will not be dropped.
0x0058 <Device Name>	Dropleless 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.

4.1.13 Receive Side Scaling (RSS)

4.1.13.1 System Requirements

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

4.1.13.2 Using RSS

Mellanox WinOF Rev 5.50 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 114.

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.

4.1.14 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.

4.1.15 VXLAN

VXLAN technology provides scalability and security challenges solutions. It requires extension of the traditional stateless offloads to avoid performance drop. ConnectX®-3 Pro adapter cards offer stateless offloads for a VXLAN packet, similar to the ones offered to non-encapsulated packets. VXLAN protocol encapsulates its packets using outer UDP header.

ConnectX®-3 Pro supports offloading of tasks related to VXLAN packet processing, such as TCP header checksum and VMQ (i.e.: directing incoming VXLAN packets to the appropriate VM queue).

Due to hardware limitation, on a dual-port adapter, VXLAN offload service cannot be provided simultaneously for both Ethernet ports if they are not using the same UDP port for VXLAN tunneling.

4.1.16 Threaded DPC

A threaded DPC is a DPC that the system executes at IRQL = PASSIVE_LEVEL. An ordinary DPC preempts the execution of all threads, and cannot be preempted by a thread or by another DPC. If the system has a large number of ordinary DPCs queued, or if one of those DPCs runs for a long period time, every thread will remain paused for an arbitrarily long period of time. Thus, each ordinary DPC increases the system latency, which can damage the performance of time-sensitive applications, such as audio or video playback.

Conversely, a threaded DPC can be preempted by an ordinary DPC, but not by other threads. Therefore, the user should use threaded DPCs rather than ordinary DPCs, unless a particular DPC must not be preempted, even by another DPC.

For more information, please refer to Introduction to Threaded DPCs.

4.1.16.1 Registry Configuration

Mlx4_bus Registry Parameters

To enable or disable this feature in the driver, set the below registry key.

Location:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\mlx4_bus\Parameters

SR-IOV Registry Keys

Key Name	Key Type	Values	Notes
ThreadDpcEnabled	DWORD	0 = Disabled 1 = Enabled	If the registry key *doesn't* exist, driver will set ThreadDpc as enabled for *Azure* packages.

4.2 InfiniBand Network

This section describes InfiniBand network features and their configuration.

- [OpenSM - Subnet Manager](#)
- [Modifying IPoIB Configuration](#)
- [Displaying Adapter Related Information](#)
- [Multiple Interfaces over Non-Default PKeys Support](#)
- [Teaming](#)

4.2.1 Port Configuration

For more information on port configuration, please refer to [Port Configuration](#) in the Ethernet Network section.

4.2.2 Assigning Port IP After Installation

For more information on port configuration, please refer to [Assigning Port IP After Installation](#) in the Ethernet Network section.

4.2.3 Receive Side Scaling (RSS)

For more information on port configuration, please refer to [Receive Side Scaling \(RSS\)](#) in the Ethernet Network section.

4.2.4 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 Manager is required in the fabric.

 Please use the embedded OpenSM in the WinOF package for testing purpose in a 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\Mellanox\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.

4.2.4.1 Modifying IPoIB Configuration

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

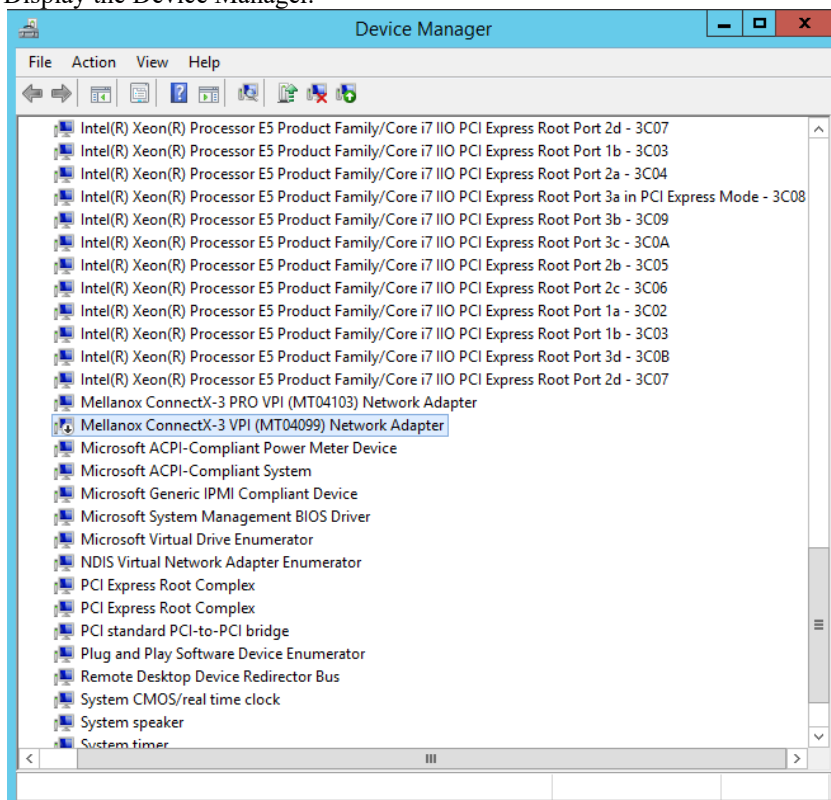
1. Open Device Manager and expand Network Adapters in the device display pane.
2. Right-click the Mellanox IPoIB Adapter entry and left-click Properties.
3. Click the Advanced tab and modify the desired properties.

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

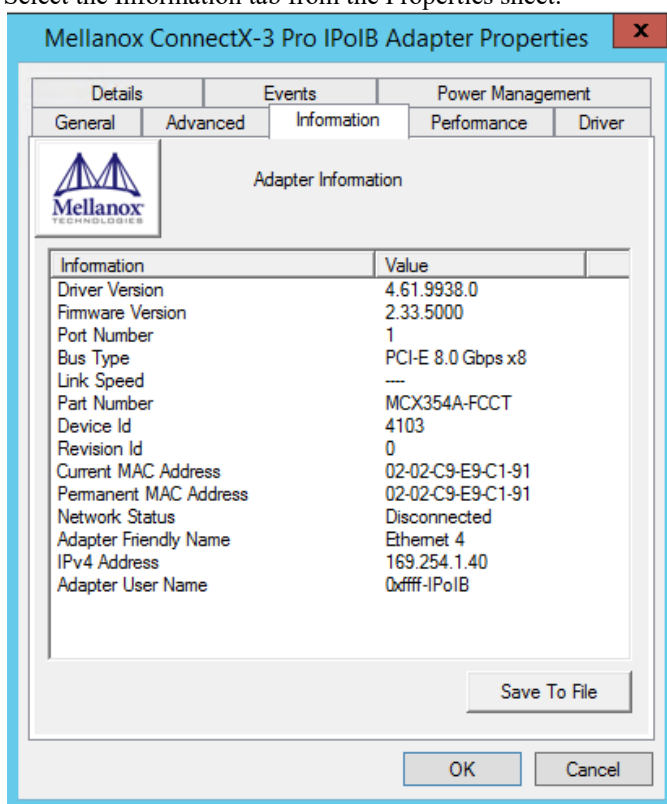
4.2.4.2 Displaying Adapter Related Information


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:

1. Display the Device Manager.



2. Select the Information tab from the Properties sheet.



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

4.2.5 Modifying IPoIB Configuration

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

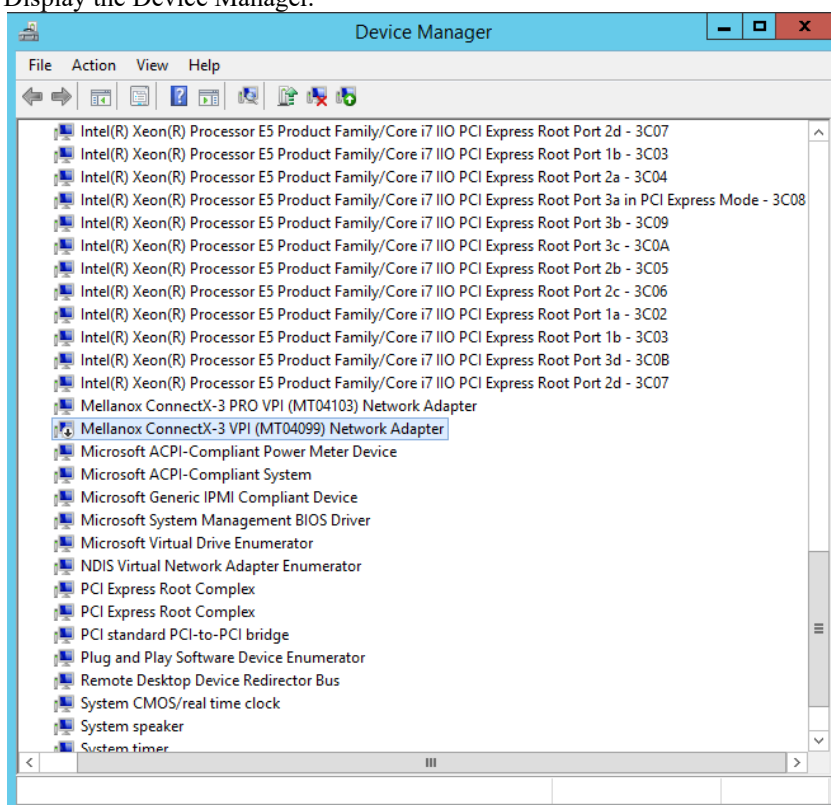
1. Open **Device Manager** and expand **Network Adapters** in the device display pane.
2. Right-click the **Mellanox IPoIB Adapter** entry and left-click **Properties**.
3. Click the **Advanced** tab and modify the desired properties.

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

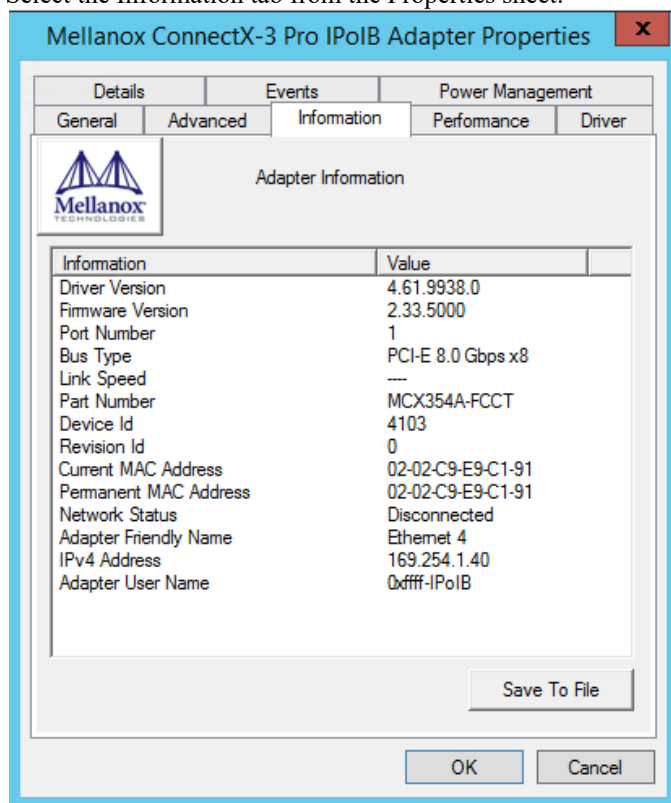
4.2.6 Displaying Adapter Related Information


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:

1. Display the Device Manager.



2. Select the Information tab from the Properties sheet.



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

4.2.7 Multiple Interfaces over Non-Default PKeys Support

4.2.7.1 System Requirements


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

4.2.7.2 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:

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*.
2. Create a new interface using the `part_man` tool.
For further details please refer to [part_man - Virtual IPoIB Port Creation Utility](#).
3. Assign Port IPs to the new interfaces.
For further details please refer to [Assigning Port IP After Installation](#) in the Ethernet Network section.

 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:

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*.
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:

1. Create a new interface using the `part_man` tool.
For further details please refer to [part_man - Virtual IPoIB Port Creation Utility](#).
2. Assign Port IPs to the new interfaces.
For further details please refer to [Assigning Port IP After Installation](#) in the Ethernet Network section.

 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:

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:

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*.
2. Map the physical PKey table to the virtual PKey table used by the VM in the following way:
 - a. Map the physical Pkey index you would like to use for the physical port to index 0 in the virtual Pkey table.
 - b. 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:

1. 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:

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.
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*.
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” in *Mellanox OFED for Linux User Manual*.
4. Enable the drivers you disabled in Windows OS.

4.2.8 Teaming

Windows Server 2012, 2012 R2 and 2016 support teaming as part of the operating system for Ethernet interfaces. However, teaming is not supported for InfiniBand adapters when using these OSes. Teaming for IPoIB is supported only when using the WinOF driver.

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

4.2.8.1 System Requirements

IPoIB teaming is supported in all operating systems supported by WinOF

4.2.8.2 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.

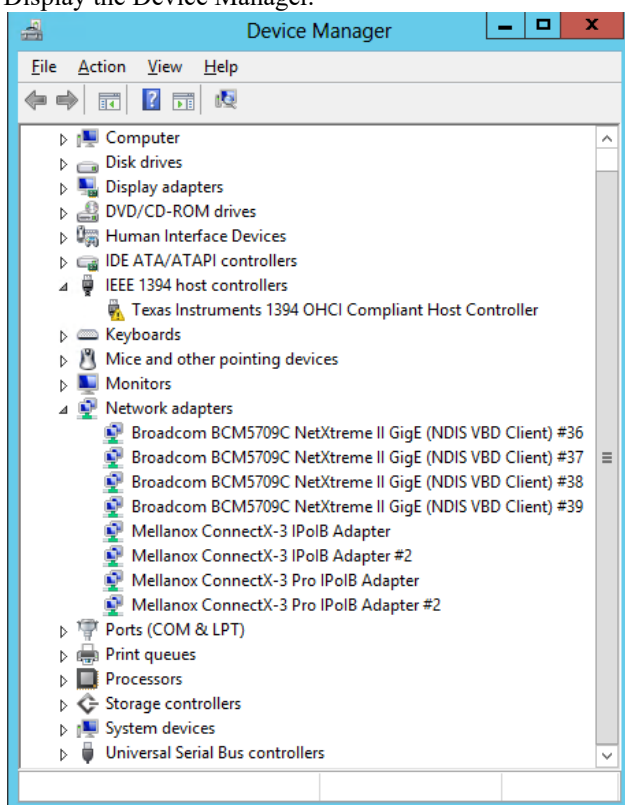
Note: For InfiniBand, the only teaming mode supported is failover.

4.2.8.3 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.

1. Display the Device Manager.



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.

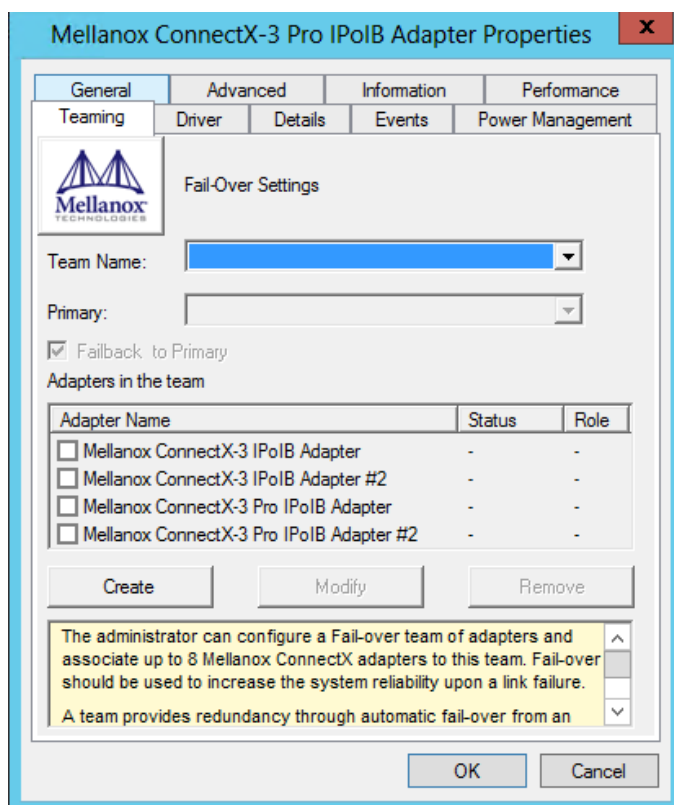
⚠ It is not recommended to open the Properties window of more than one adapter simultaneously.

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:

1. Click Create.
2. Enter a (unique) team name.
3. Select the adapters to be included in the team.
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.
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

General | Advanced | Information | Performance

Teaming | Driver | Details | Events | Power Management

Fail-Over Settings

Team Name:

Primary:

☒ Failback to Primary

Adapters in the team

Adapter Name	Status	Role
<input type="checkbox"/> Mellanox ConnectX-3 IPoIB Adapter	-	-
<input type="checkbox"/> Mellanox ConnectX-3 IPoIB Adapter #2	-	-
<input type="checkbox"/> Mellanox ConnectX-3 Pro IPoIB Adapter	-	-
<input type="checkbox"/> Mellanox ConnectX-3 Pro IPoIB Adapter #2	-	-

Create Modify Remove

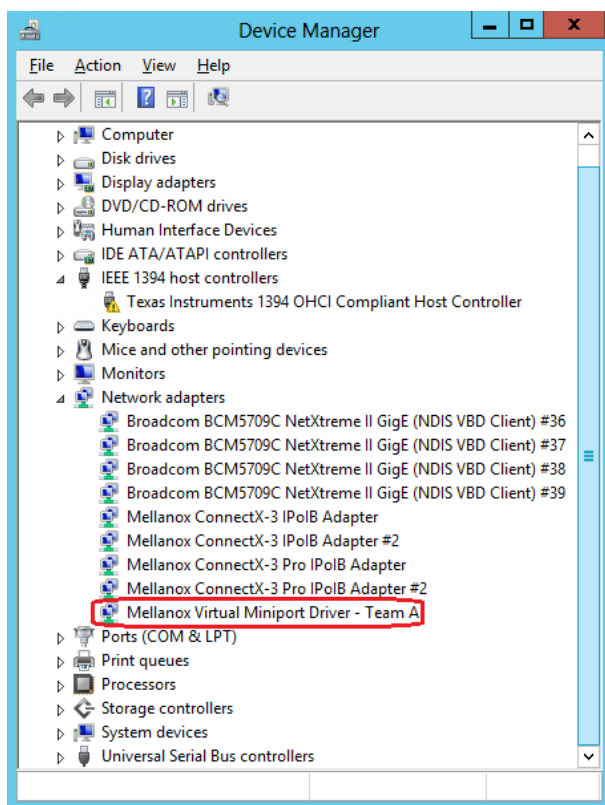
The administrator can configure a Fail-over team of adapters and associate up to 8 Mellanox ConnectX adapters to this team. Fail-over should be used to increase the system reliability upon a link failure.

A team provides redundancy through automatic fail-over from an

OK Cancel

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>



To modify an existing team, perform the following:

1. Select the desired team and click Modify
2. Modify the team name and/or the participating adapters
3. 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:

1. Each adapter that participates in a team has two properties:
 - Status: Connected/Disconnected/Disabled
 - Role: Active or Backup
2. 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.
3. 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.

4.3 Management

4.3.1 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:

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 policy?
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): y
```

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\FFFFFFFFFFFFFFFFF00
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\FFFFFFFFFFFFFFFFF00
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&FFFFFFFFFFFFFFFFF00
Name               : HCA 2
Source             : 3
SystemName         : WIN-CQM7PRQFHUO
SriovEnable        : False
SriovPort1NumVFs   : 16
SriovPort2NumVFs   : 0
SriovPortMode      : 0
PSComputerName     :

PS C:\Users\Administrator>
```

4.4 Storage Protocols

4.4.1 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.

4.4.1.1 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

4.4.2 SMB Configuration Verification

4.4.2.1 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
```

4.4.2.2 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:

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


Note: The NETSTAT command confirms if the File Server is listening on the RDMA interfaces.

4.4.2.3 Verifying SMB Connection

To verify the SMB connection on the SMB client:

1. Copy the large file to create a new session with the SMB Server.
2. Open a PowerShell window while the copy is ongoing.
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.

4.4.2.4 Verifying SMB Events that Confirm RDMA Connection

To confirm RDMA connection, verify the SMB events:

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"
```

4.5 Virtualization

This section describes virtualization features and their configuration.

- [Hyper-V with VMQ](#)
- [Network Virtualization using Generic Routing Encapsulation \(NVGRE\)](#)
- [Single Root I/O Virtualization \(SR-IOV\)](#)
 - [Configuring Operating Systems](#)
- [Virtual Machine Multiple Queue \(VMMQ\)](#)
- [PacketDirect Provider Interface](#)

4.5.1 Hyper-V with VMQ

4.5.1.1 System Requirements

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

4.5.1.2 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:

1. Open Hyper-V Manager.
2. Right-click the desired Virtual Machine (VM), and left-click Settings in the pop-up menu.
3. In the Settings window, under the relevant network adapter, select "Hardware Acceleration".
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:

1. Enable VMQ on a specific VM: Set-VMNetworkAdapter <VM Name> -VmqWeight 100
2. Disable VMQ on a specific VM: Set-VMNetworkAdapter <VM Name> -VmqWeight 0

4.5.2 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.

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

For further information, please refer to the Microsoft's "Network Virtualization using Generic Routing Encapsulation (NVGRE) Task Offload" document at <https://docs.microsoft.com/en-us/windows-hardware/drivers/network/network-virtualization-using-generic-routing-encapsulation--nvgre--task-offload>

4.5.2.1 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:

1. Open the Device Manager.
2. Go to the Network adapters.
3. Right click 'Properties' on Mellanox ConnectX®-3 Pro Ethernet Adapter card.
4. Go to Advanced tab.
5. Choose the 'Encapsulate Task Offload' option.
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 significantly reduces performance.

4.5.2.1.1 Configuring the NVGRE using PowerShell


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

For further information of how to configure NVGRE using PowerShell, please refer to Microsoft's "Step-by-Step: Hyper-V Network Virtualization" blog at <https://blogs.technet.microsoft.com/keithmayer/2012/10/08/step-by-step-hyper-v-network-virtualization-31-days-of-favorite-features-in-winserv-2012-part-8-of-31/>

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.

4.5.3 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 additional 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 [Enabling SR-IOV in Mellanox WinOF Package \(Ethernet SR-IOV Only\)](#)).

4.5.3.1 SR-IOV Ethernet over Hyper-V

4.5.3.1.1 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

4.5.3.1.2 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

4.5.3.2 SR-IOV InfiniBand over KVM

4.5.3.2.1 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

4.5.3.2.2 Feature Limitations (Compared to Native InfiniBand)

- OpenSM and Infiniband Fabric Diagnostic Utilities listed in Table 51, “Diagnostic Utilities,” on page 184 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.

4.5.3.3 Configuring SR-IOV Host Machines

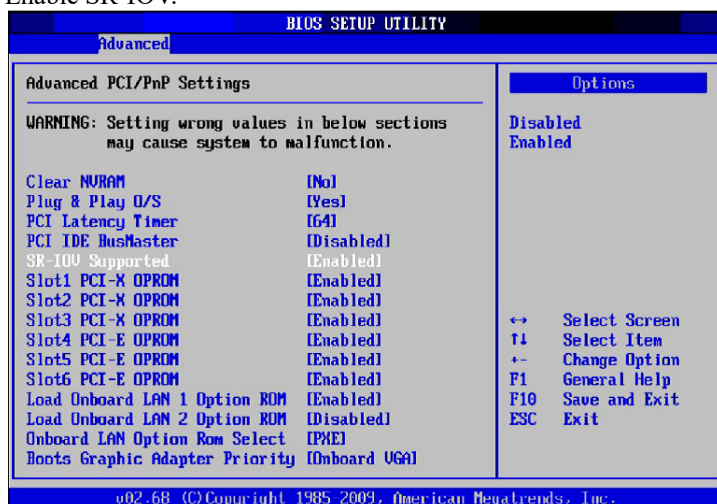
The following are the necessary steps for configuring host machines.

4.5.3.3.1 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:

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



- b. Enable "Intel Virtualization Technology" Support.

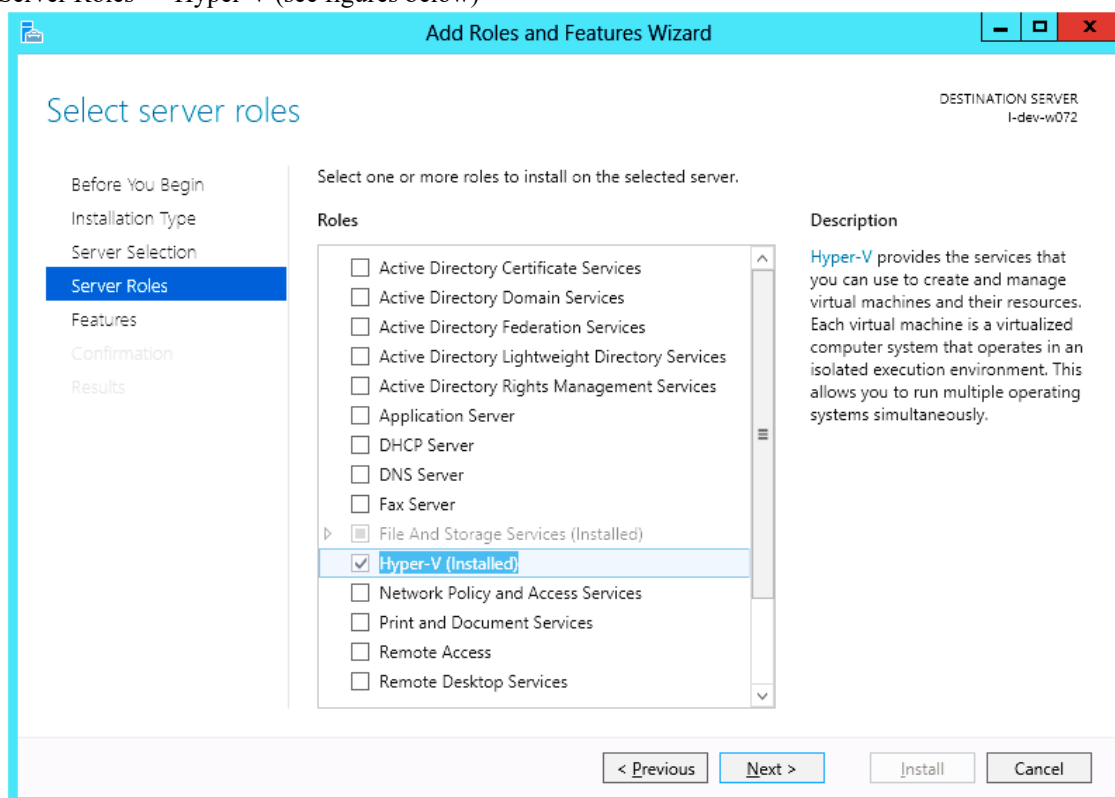


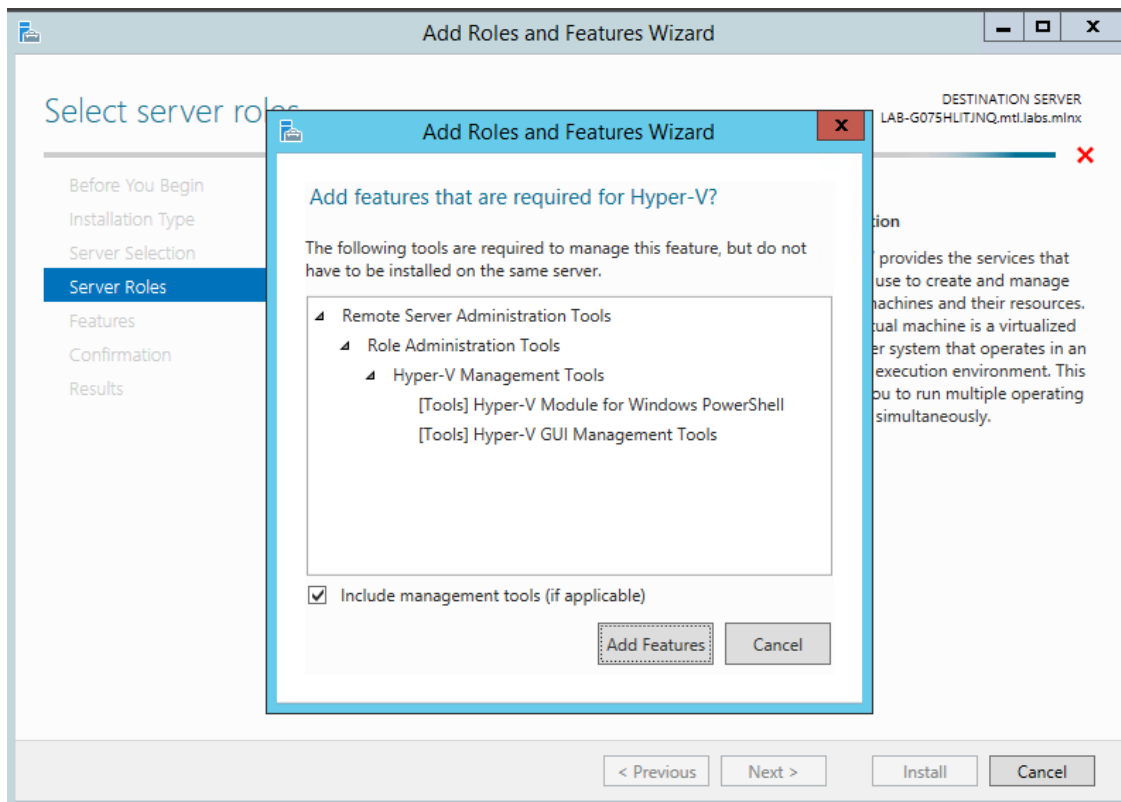
For further details, please refer to the vendor's website.

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

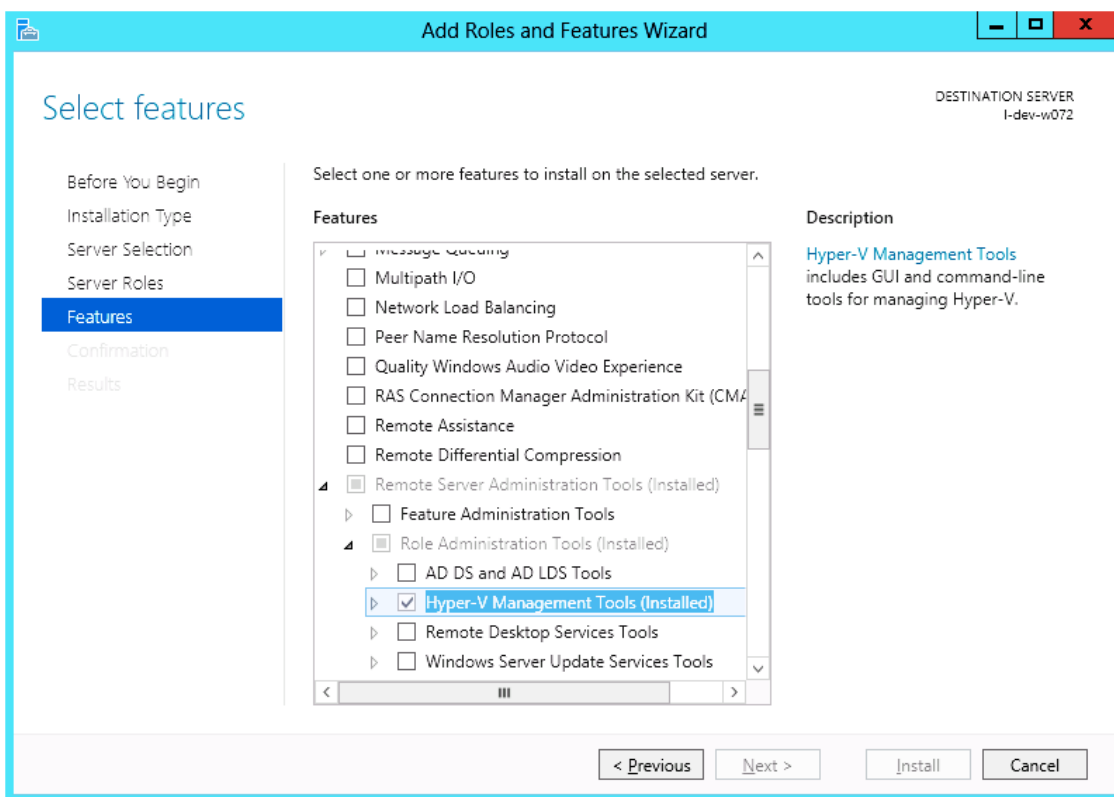
To install Hypervisor Operating System:

1. Install Windows Server 2012 R2 and above.
2. Install Hyper-V role:
 - a. Go to: Server Manager → Manage → Add Roles and Features and set the following:
 - b. Installation Type → Role-based or Feature-based Installation
 - c. Server Selection → Select a server from the server pool
 - d. Server Roles → Hyper-V (see figures below)

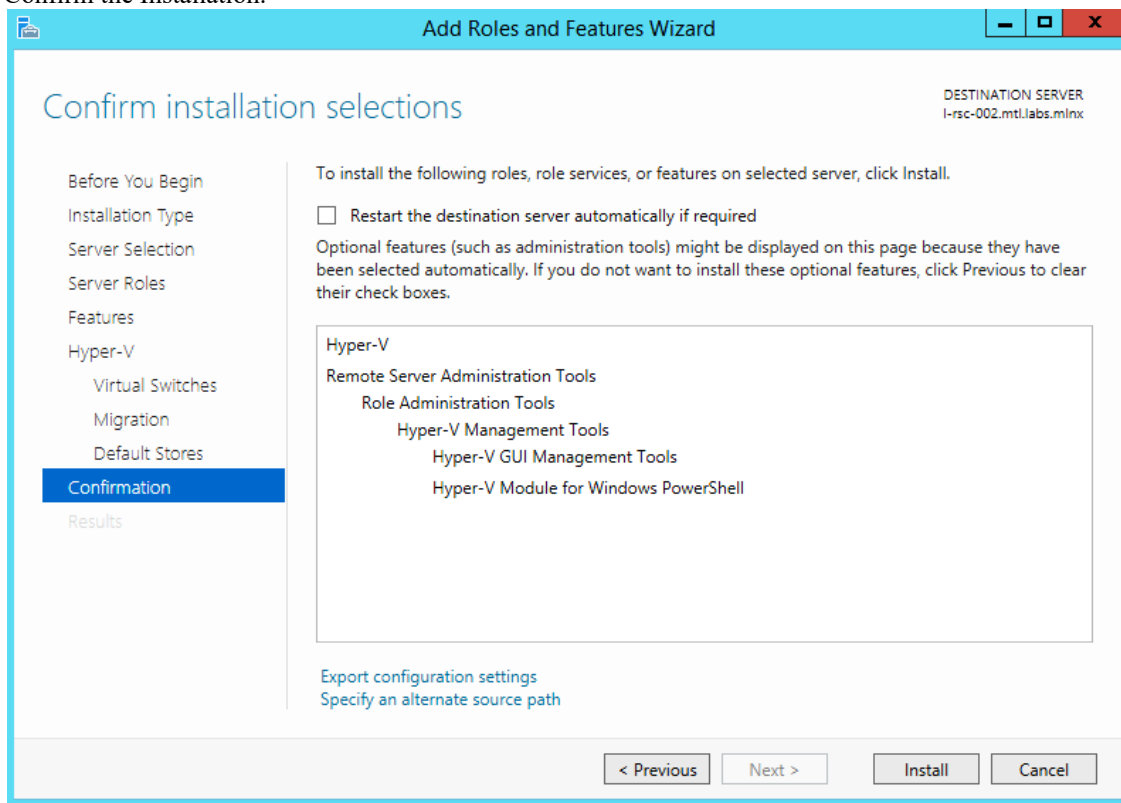




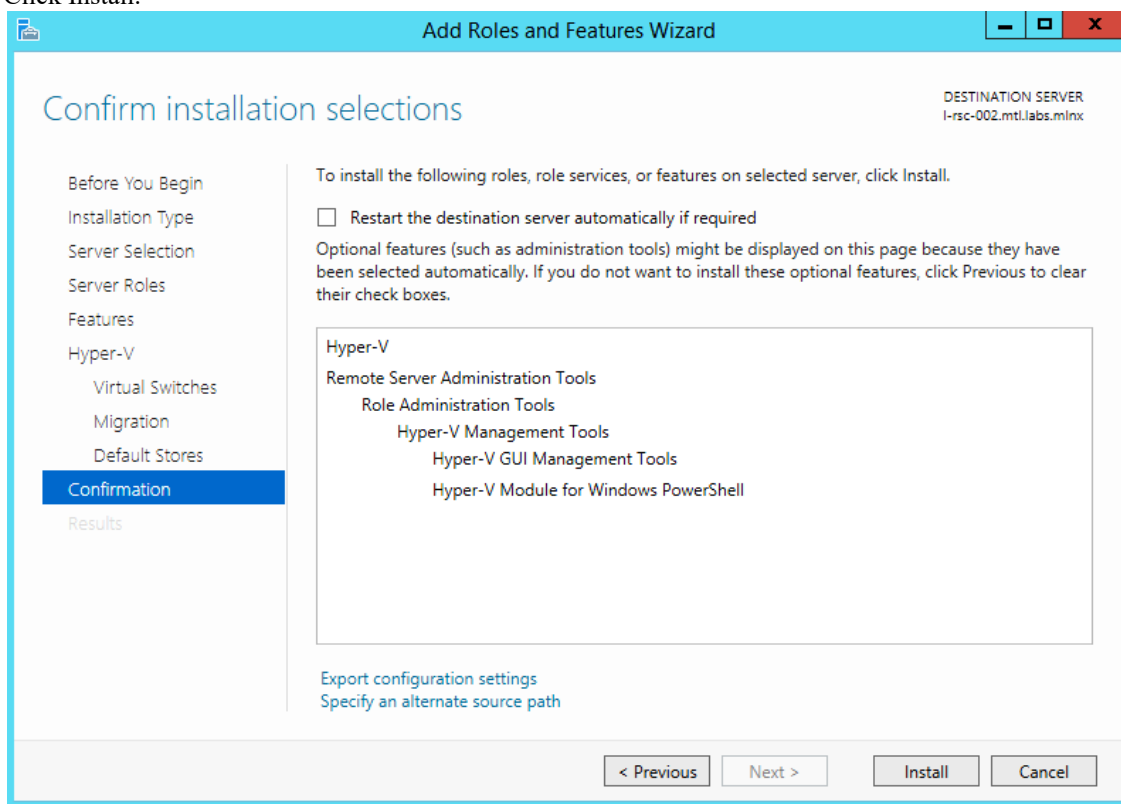
3. Install Hyper-V Management Tools. Features → Remote Server Administration Tools → Role Administration Tools → Hyper-V Administration Tool



4. Confirm the Installation.



5. Click Install.



6. Reboot the system.

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

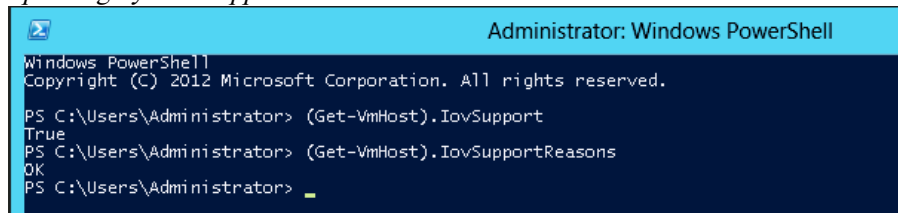
To verify that the system is properly configured for SR-IOV:

1. Go to: Start → Windows Powershell.
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.

Operating System Supports SR-IOV

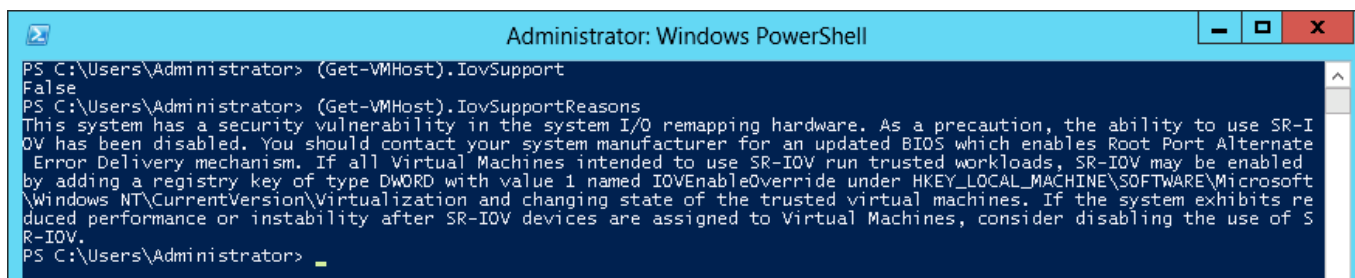


```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2012 Microsoft Corporation. All rights reserved.

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.

SR-IOV Support



```
Administrator: Windows PowerShell
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-IOV 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 reduced performance or instability after SR-IOV devices are assigned to Virtual Machines, consider disabling the use of SR-IOV.
PS C:\Users\Administrator>
```

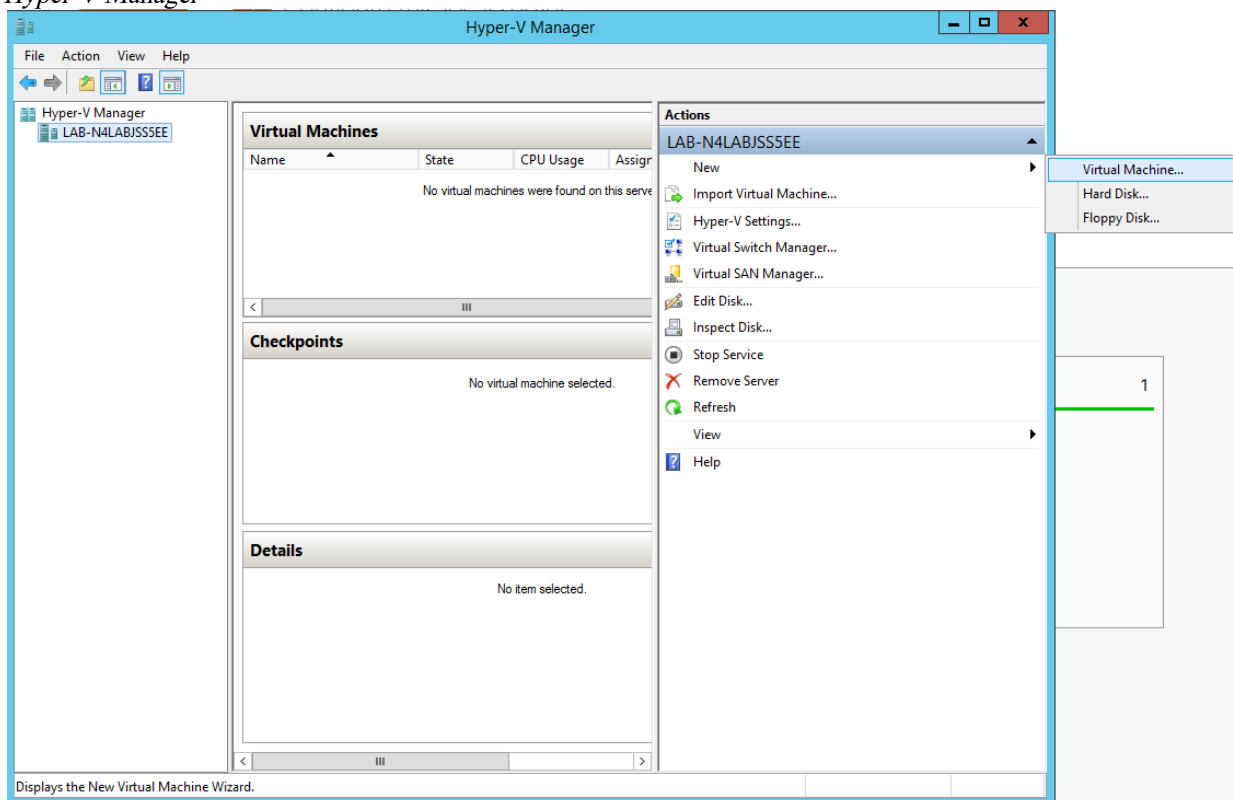
3. Reboot
4. Verify the system is configured correctly for SR-IOV as described in Steps 1/2.

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

To create a virtual machine

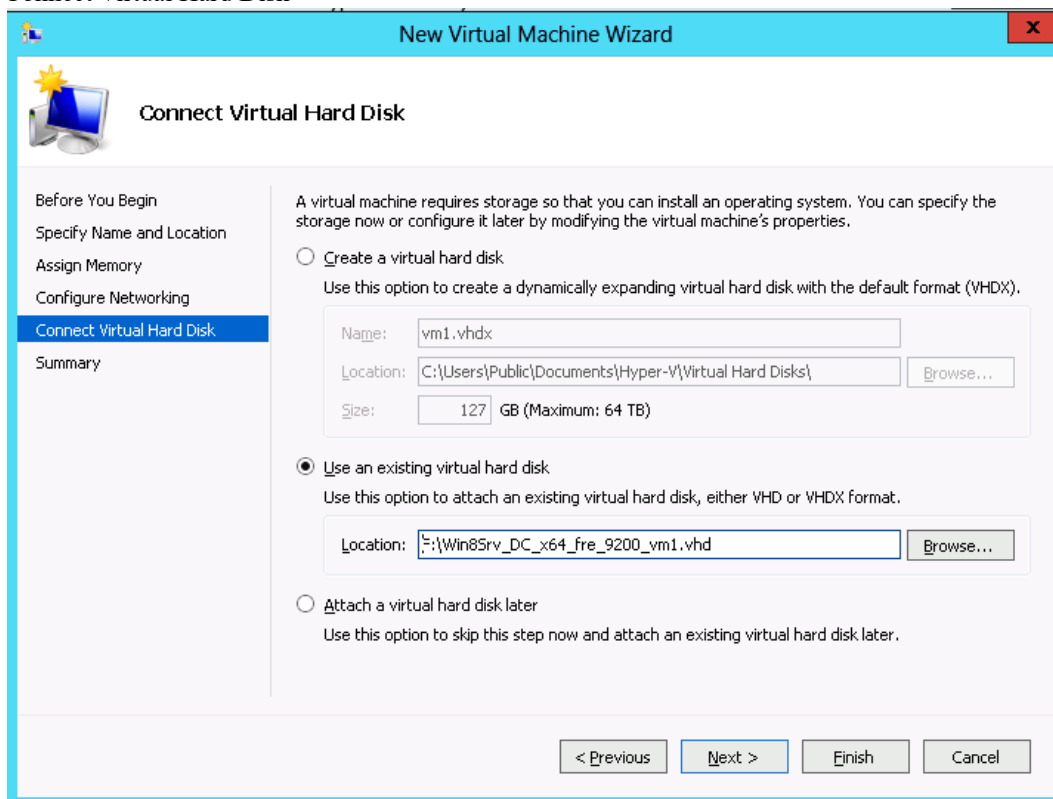
1. Go to: Server Manager → Tools → Hyper-V Manager.
2. Go to: New → Virtual Machine and set the following:
 - Name: <name>
 - Startup memory: 4096 MB
 - Connection: Not Connected

Hyper-V Manager



3. Connect the virtual hard disk in the New Virtual Machine Wizard.
4. Go to Connect Virtual Hard Disk → Use an existing virtual hard disk.
5. Select the location of the vhd file.

Connect Virtual Hard Disk



4.5.3.4 Configuring Mellanox Network Adapter for SR-IOV

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

4.5.3.4.1 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 tool. For older firmware versions, use the flint tool.

4.5.3.4.1.1 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.

1. Download MFT for Windows. www.mellanox.com → Products → Software → Firmware Tools.
2. Get the device ID (look for the “_pciconf” string in the output).

```
> mst status
```

Example:

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

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
LINK_TYPE_P1  N/A
LINK_TYPE_P2  N/A
```

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
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.
```

5. Reboot the machine. (After the reboot, continue to [Enabling SR-IOV in Mellanox WinOF Package \(Ethernet SR-IOV Only\)](#).)

4.5.3.4.1.2 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.

Recommended Values

Parameter	Recommended Values
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 -wimage <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.

4.5.3.4.2 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 [Port Configuration](#).

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

Step 3. Set the Execution Policy specified in [PowerShell Configuration](#).

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.

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

Note: "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.

Example:

```
Confirm
Are you sure you want to perform this action?
Performing the operation "SetValue" on target "MLNX_PCIDeviceSriovSettingData:
MLNX_PCIDeviceSriovSettingData
'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.

SR-IOV mode configuration parameters:

Parameter Name	Values	Descriptions
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.
SriovPort1NumVFs SriovPort2NumVFs	16=(default)	SriovPort<i>NumVFs specifies the number of VFs that are allowed per port. This is the number of VFs the bus driver will open when working in manual mode on SRI-OV mode. Note: If the total number of VFs requested is larger than the number of VFs burnt in the 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 the 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 :

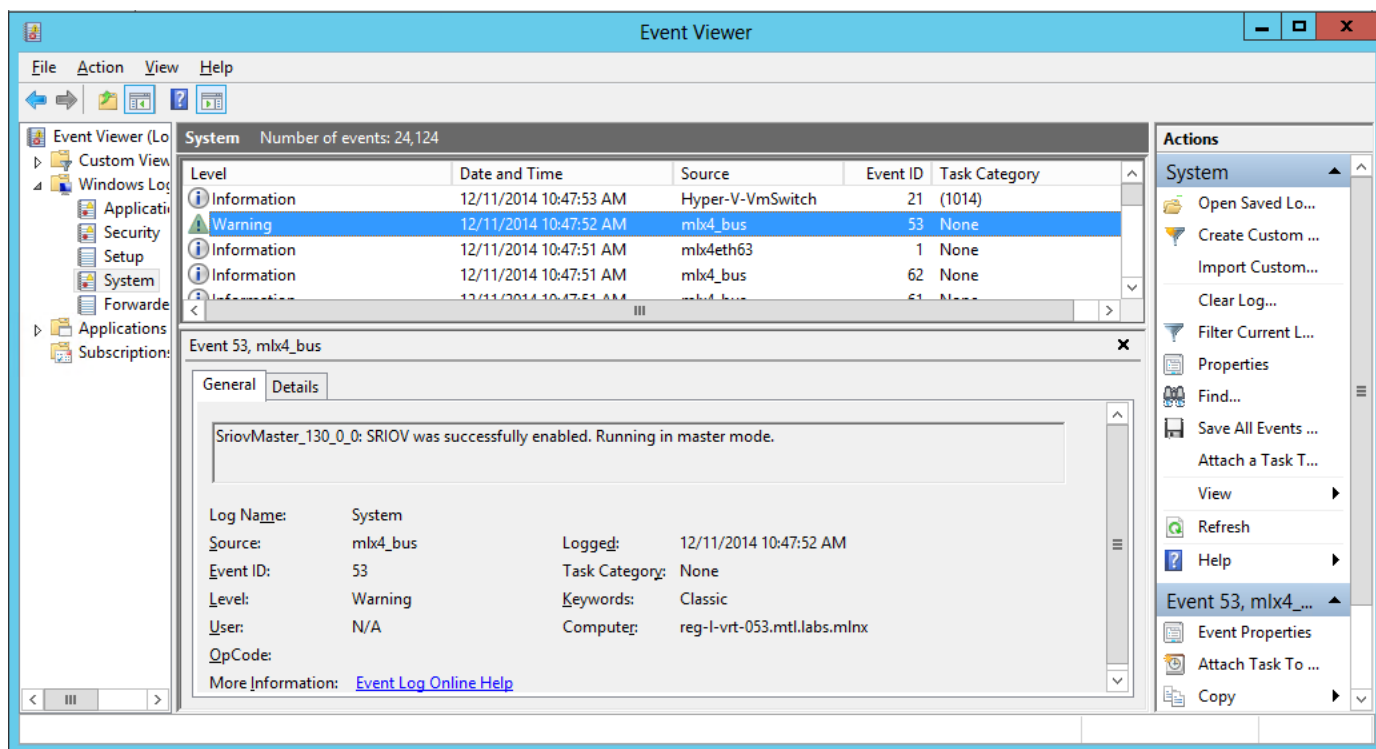
```

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.

System Event Log



4.5.3.5 Configuring Operating Systems

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

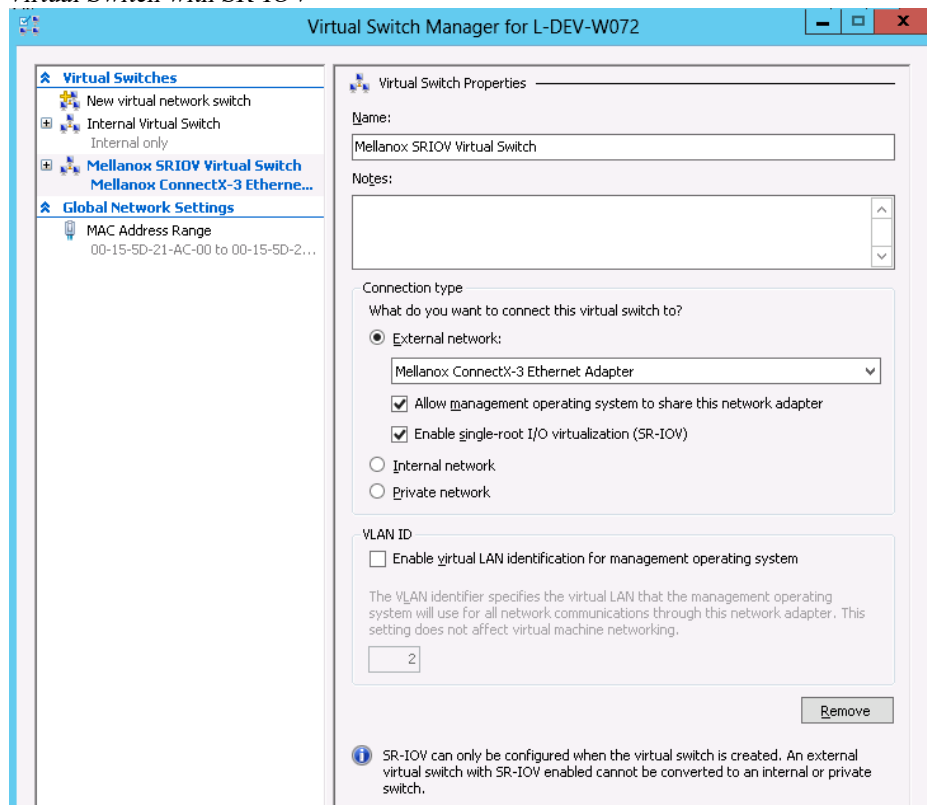
For further details on enabling/configuring SR-IOV on KVM, please refer to the section titled “Single Root IO Virtualization (SR-IOV)” in *Mellanox OFED for Linux User Manual*.

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

To configure Virtual Machine networking:

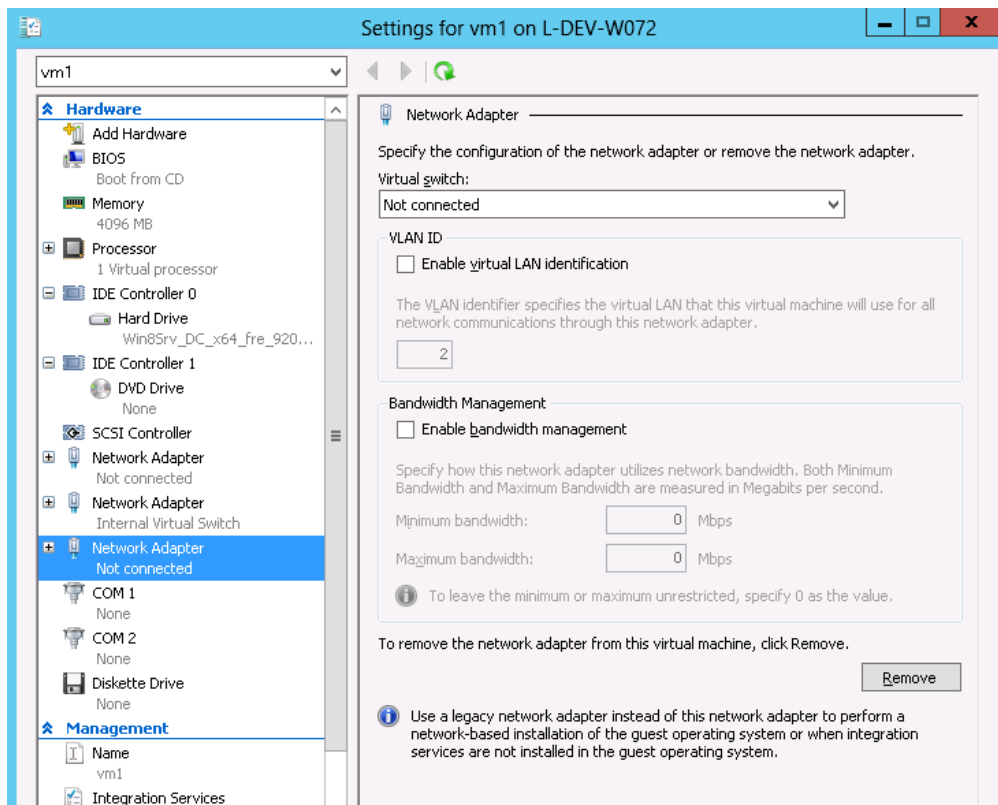
1. Create an SR-IOV-enabled Virtual Switch over Mellanox Ethernet Adapter.
Go to Start → Server Manager → Tools → Hyper-V Manager.
In the Hyper-V Manager: Actions → Virtual SwitchManager → External → Create Virtual Switch.
2. Set the following:
 - Name:
 - External network:
 - Enable single-root I/O virtualization (SR-IOV)

Virtual Switch with SR-IOV



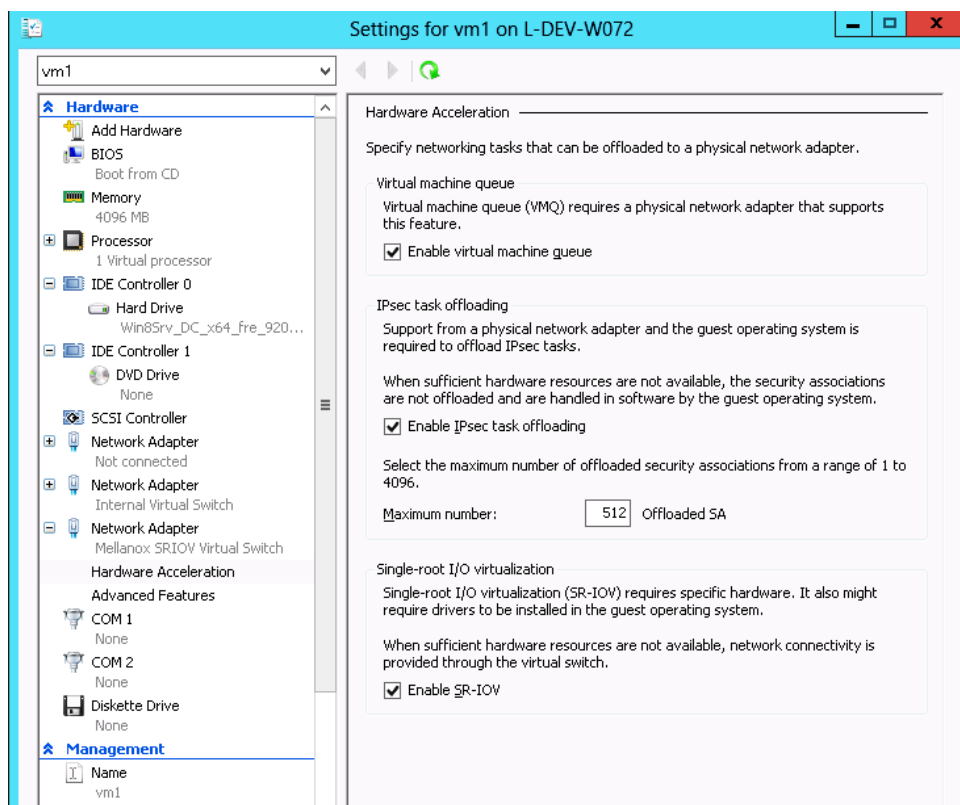
3. Click Apply.
4. Click OK.
5. Add a VMNIC connected to a Mellanox vSwitch in the VM hardware settings:
 - a. Under Actions, go to Settings → Add New Hardware → Network Adapter → OK.
 - b. In “Virtual Switch” dropdown box, choose Mellanox SR-IOV Virtual Switch.

Adding a VMNIC to a Mellanox V-switch



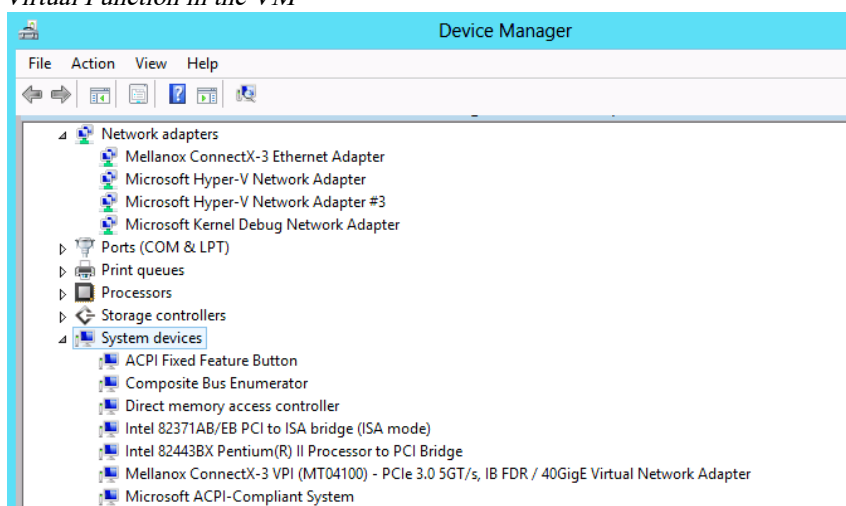
6. Enable the SR-IOV for Mellanox VMNIC:
 - a. Open VM settings Wizard.
 - b. Open the Network Adapter and choose Hardware Acceleration.
 - c. Tick the “Enable SR-IOV” option.
 - d. Click OK.

Enable SR-IOV on VMNIC



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.
8. Copy the WinOF driver package to the VM using Mellanox VMNIC IP address.
9. Install WinOF driver package on the VM.
10. Reboot the VM at the end of installation.
11. Verify that Mellanox Virtual Function appears in the device manager.

Virtual Function in the VM



⚠ 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

4.5.3.5.3 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:

Default Ethertype Values

Value	Description
0x0800	Internet Protocol version 4 (IPv4)
0x0806	Address Resolution Protocol (ARP)
0x86DD	Internet Protocol Version 6 (IPv6)

4.5.3.5.3.1 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

The group below contains the registry keys through which you can configure the feature: For more information on how to find device index nn, please refer to [Finding the Index Value of the HCA](#).

Ethertype Spoof Protection Registry Keys

Key Name	Key Type	Values	Description
VFAIlowedTxEtherTypeListEnable	REG_DWORD	0 = Disabled 1 = Enabled	Enables/disables the feature
VFAIlowedTxEtherType0	REG_DWORD	Ethertype value	The first Ethertype to allow VF to transmit
VFAIlowedTxEtherType1	REG_DWORD	Ethertype value	The second Ethertype to allow VF to transmit
VFAIlowedTxEtherType2	REG_DWORD	Ethertype value	The third Ethertype to allow VF to transmit
VFAIlowedTxEtherType3	REG_DWORD	Ethertype value	The fourth Ethertype to allow VF to transmit
VFAIlowedTxEtherType4	REG_DWORD	Ethertype value	The fifth Ethertype to allow VF to transmit

Key Name	Key Type	Values	Description
VFAIlowedTxEtherType5	REG_DWORD	Ethertype value	The sixth EtherType to allow VF to transmit
VFAIlowedTxEtherType6	REG_DWORD	Ethertype value	The seventh EtherType to allow VF to transmit
VFAIlowedTxEtherType7	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 VFAIlowedTxEtherType0..7) with value of 0x0.

4.5.4 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.

4.5.4.1 System Requirements

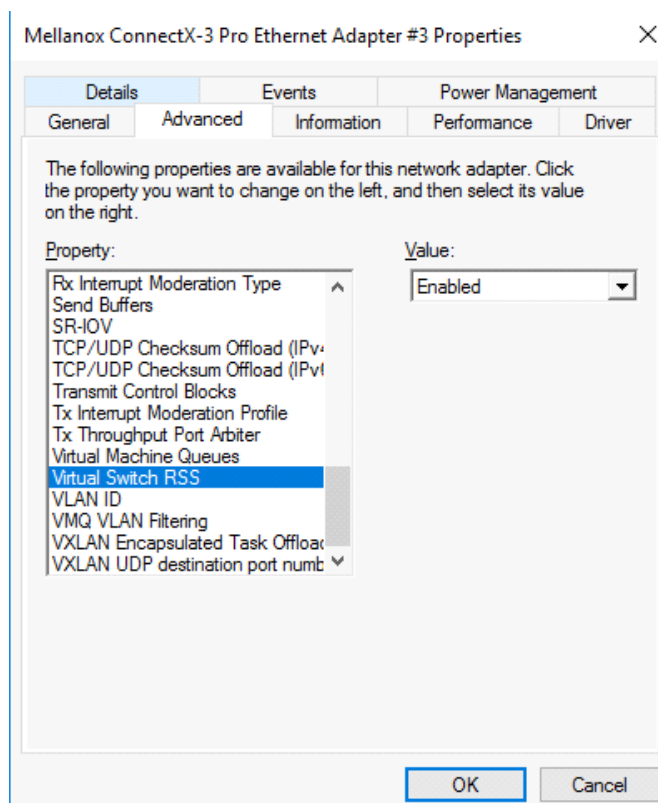
- Operating System(s): Windows Server 2016
- Mellanox ConnectX-3/ConnectX-3 Pro VPI adapter card family
- Available only for Ethernet (no IPOIB)

4.5.4.2 Enabling/Disabling VMMQ

4.5.4.2.1 On the Driver Level

To enable/disable VMMQ:

1. Go to: Display Manager → Network adapters → Mellanox ConnectX-3 Ethernet Adapter → Properties → Advanced tab → Virtual Switch Rss.



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

4.5.4.2.2 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.

4.5.4.3 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 "VMName" -name "Virtual Adapter Name" -VmmqQueuePairs <number>
```

⚠ 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.

4.5.5 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.

4.5.5.1 System Requirements

- Hypervisor OS: Windows Server 2016
- Virtual Machine (VM) OS: Windows Server 2012 and above
- Mellanox ConnectX-3 Pro Adapter Card family
- Mellanox WinOF 5.25 or higher
- Firmware version: 2.36.5150 or higher

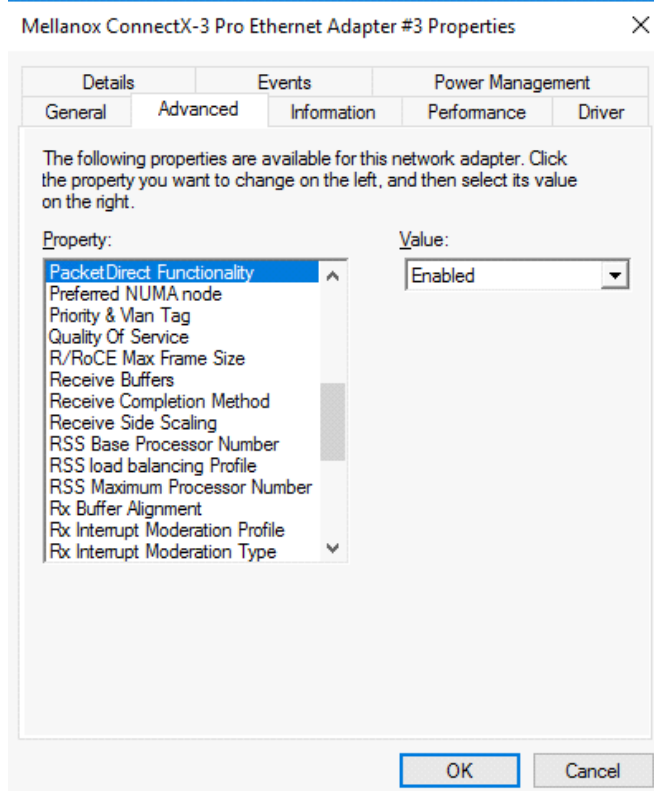
4.5.5.2 Using PacketDirect for VM

To allow a VM to send/receive traffic in PacketDirect mode:

1. Enable PacketDirect:
 - On the Ethernet adapter.

```
PS $ Enable-NetAdapterPacketDirect -Name <EthInterfaceName>
```

- In the Device Manager:



2. Create a vSwitch with PacketDirect enabled.

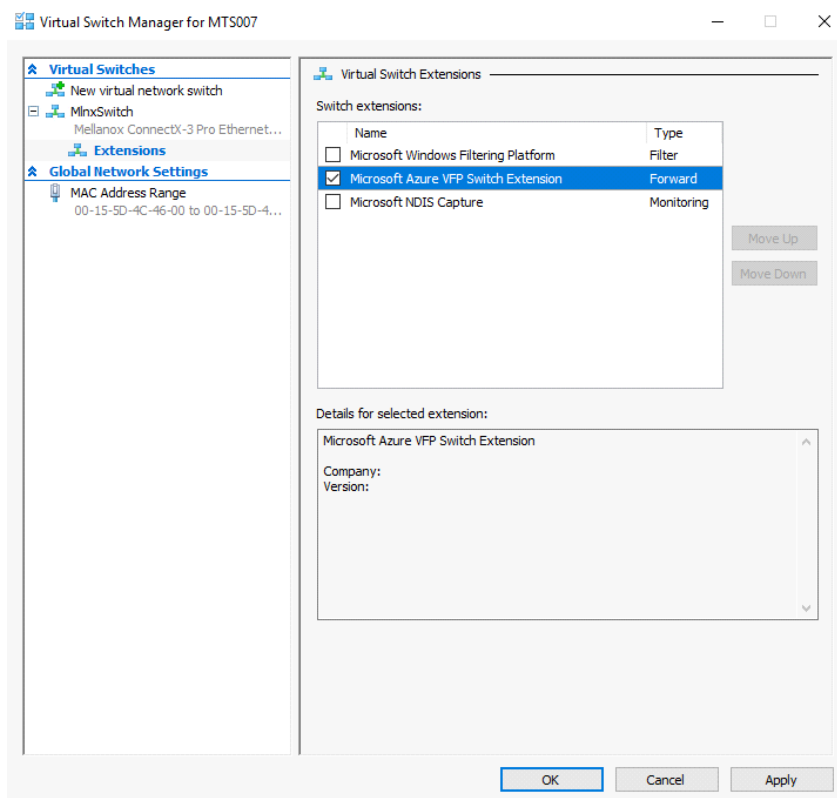
```
PS $ New-VMSwitch <vSwitchName> -NetAdapterName <EthInterfaceName> -EnablePacketDirect $true -AllowManagementOS $true
```

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...



4. Shut down the VM.

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

5. Add a virtual network adapter for the VM.

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

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:

- 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!


```

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.

4.6 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:

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.

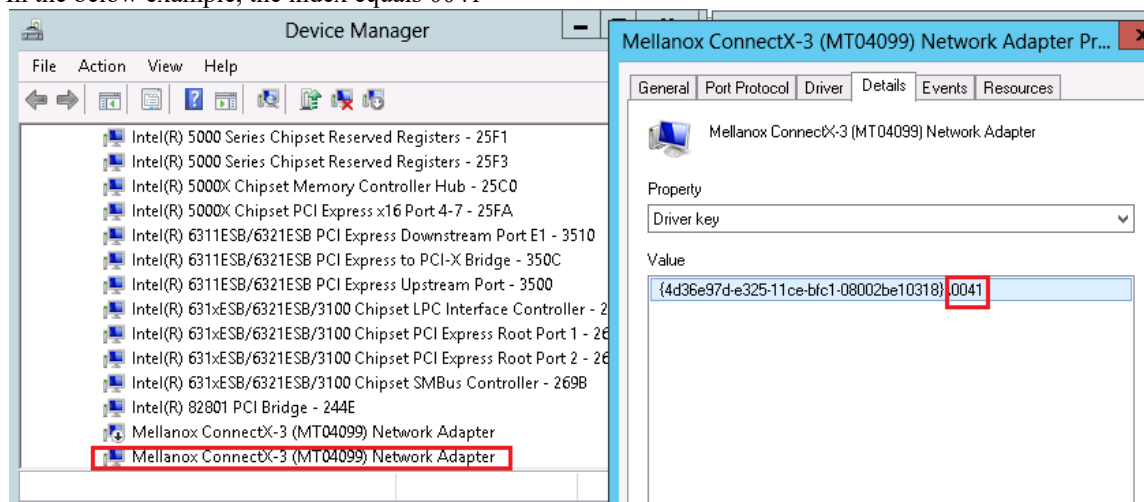
Any registry key that starts with an asterisk ("*") is a well-known registry key. For more details regarding the registries, please refer to: [http://msdn.microsoft.com/en-us/library/ff570865\(v=VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ff570865(v=VS.85).aspx)

4.6.1 Finding the Index Value of the HCA

To find the nn value of your HCA from the Device Manager please perform the following steps:

1. Open Device Manager, and go to System devices.
2. Right click on a Mellanox -ConnectX® card → Properties.
3. Go to Details tab.
4. Select the Driver key, and obtain the nn number.

In the below example, the index equals 0041

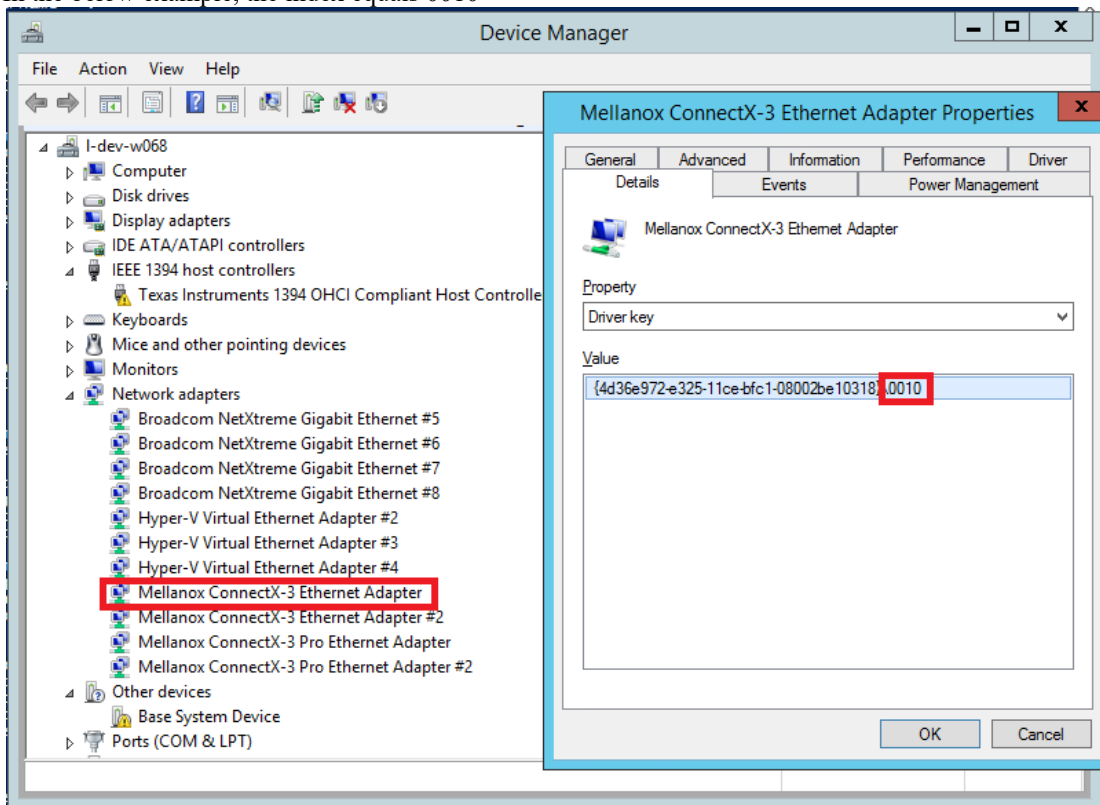


4.6.2 Finding the Index Value of the Network Interface

To find the index value of your Network Interface from the Device Manager please perform the following steps:

1. Open Device Manager, and go to Network Adapters.
2. Right click → Properties on Mellanox Connect-X® Ethernet Adapter.
3. Go to Details tab.

4. Select the Driver key, and obtain the nn number.
In the below example, the index equals 0010



4.6.3 Basic Registry Keys

This group contains the registry keys that control the basic operations of the NIC.

Value Name	Default Value	Description
*JumboPacket	eth:1514 IPoIB:4096	<p>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.</p> <p>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.</p> <p>The valid values are:</p> <ul style="list-style-type: none"> Ethernet: 600 up to 9600 IPoIB: 1500 up to 4092 <p>Notes:</p> <ul style="list-style-type: none"> 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	<p>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.</p> <p>In case of lack of received buffers (dropped packets or out of order received packets), you can increase the number of received buffers.</p> <p>The valid values are 256 up to 4096.</p>
*TransmitBuffers	eth:2048 IPoIB:2048	<p>The number of packets each ring sends. Increasing this value can enhance transmission performance, but also consumes system memory.</p> <p>The valid values are 256 up to 4096.</p>
*SpeedDuplex	7	<p>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 meaning 10Gbps and Full Duplex.</p> <p>Note: Default value should not be modified.</p>
MaxNumOfMCList	eth:128 IPoIB:128	<p>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.</p> <p>The valid values are 64 up to 1024.</p> <p>Note: This registry value is not exposed via the UI.</p>

Value Name	Default Value	Description
*QOS	eth:1	<p>Enables the NDIS Quality of Service (QoS) The valid values are:</p> <ul style="list-style-type: none"> 1: enable 0: disable <p>Notes:</p> <ul style="list-style-type: none"> 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	<p>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.</p> <p>The valid values are:</p> <ul style="list-style-type: none"> 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.
TxIntModerationProfile	eth:1 IPoIB:1	<p>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:</p> <ul style="list-style-type: none"> 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.

4.6.4 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.

Value Name	Default Value	Description
*LsoV1IPv4	1	Large Send Offload Version 1 (IPv4). The valid values are: <ul style="list-style-type: none"> • 0: disable • 1: enable
*LsoV2IPv4	1	Large Send Offload Version 2 (IPv4). The valid values are: <ul style="list-style-type: none"> • 0: disable • 1: enable
*LsoV2IPv6	1	Large Send Offload Version 2 (IPv6). The valid values are: <ul style="list-style-type: none"> • 0: disable • 1: enable
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: <ul style="list-style-type: none"> • 0: disable • 1: enable Note: This registry key is not exposed to the user via the UI.

Value Name	Default Value	Description
LSOIpOptions	eth:1 IPoIB:1	<p>Enables its NIC to segment a large TCP packet whose IP header contains IP options.</p> <p>The valid values are:</p> <ul style="list-style-type: none"> • 0: disable • 1: enable <p>Note: This registry key is not exposed to the user via the UI.</p>
*IPChecksumOffloadIPv4	eth:3 IPoIB:3	<p>Specifies whether the device performs the calculation of IPv4 checksums.</p> <p>The valid values are:</p> <ul style="list-style-type: none"> • 0: (disable) • 1: (Tx Enable) • 2: (Rx Enable) • 3: (Tx and Rx enable)
*TCPUDPChecksumOffloadIPv4	eth:3 IPoIB:3	<p>Specifies whether the device performs the calculation of TCP or UDP checksum over IPv4.</p> <p>The valid values are:</p> <ul style="list-style-type: none"> • 0: (disable) • 1: (Tx Enable) • 2: (Rx Enable) • 3: (Tx and Rx enable)
*TCPUDPChecksumOffloadIPv6	eth:3 IPoIB:3	<p>Specifies whether the device performs the calculation of TCP or UDP checksum over IPv6.</p> <p>The valid values are:</p> <ul style="list-style-type: none"> • 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	<p>Virtual Machine Multiple Queue (VMMQ) HW Offload</p> <p>The valid values are:</p> <ul style="list-style-type: none"> • 0: disable • 1: enable

4.6.5 Performance Registry Keys

This group of registry keys configures parameters that can improve adapter performance.

Value Name	Default Value	Description
RecvCompletion Method	eth:1 IPoIB:1	<p>Sets the completion methods of the receive packets, and it affects network throughput and CPU utilization.</p> <p>The supported methods are:</p> <ul style="list-style-type: none"> • 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. <p>The valid values are:</p> <ul style="list-style-type: none"> • 0: polling • 1: adaptive
*InterruptModeration	eth:1 IPoIB:1	<p>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.</p> <p>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.</p> <p>The valid values are:</p> <ul style="list-style-type: none"> • 0: disable • 1: enable
RxIntModeration	eth:2 IPoIB:2	<p>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.</p> <p>The valid values are:</p> <ul style="list-style-type: none"> • 1: static • 2: adaptive <p>The interrupt moderation count and time are configured dynamically, based on traffic types and rate.</p>
pkt_rate_low	eth: 150000 IPoIB: 150000	<p>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.</p> <p>Note: This registry value is not exposed via the UI.</p>

Value Name	Default Value	Description
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.
*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: <ul style="list-style-type: none"> • 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.
TxHashDistribution	3	Sets the algorithm which is used to distribute the send-packets on different send rings. The adapter uses 3 methods: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • 1: size • 2: hash • 3: hash and size Note: This registry value is not exposed via the UI.
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: <ul style="list-style-type: none"> • 0: disable • 1: enable Note: This registry value is not exposed via the UI.

Value Name	Default Value	Description
ReturnPacketThreshold	eth:341 IPoIB:341	<p>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.</p> <p>When the value is set to 0, the adapter uses 2/3 of the received ring size.</p> <p>The valid values are: 0 to 4096.</p> <p>Note: This registry value is not exposed via the UI.</p>
NumTcb	eth:16 IPoIB:16	<p>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.</p> <p>The valid values are 1 up to 64.</p> <p>Note: This registry value is not exposed via the UI.</p>
ThreadPoll	eth:10000 IPoIB:10000	<p>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.</p> <p>The valid values are 0 up to 200000.</p> <p>Note: This registry value is not exposed via the UI.</p>
AverageFactor	eth:16 IPoIB:16	<p>The weight of the last polling in the decision whether to continue the polling or give up when using polling completion method for receiving.</p> <p>The valid values are 0 up to 256.</p> <p>Note: This registry value is not exposed via the UI.</p>
AveragePollThreshold	eth:10 IPoIB:10	<p>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.</p> <p>The valid values are 0 up to 1000.</p> <p>Note: This registry value is not exposed via the UI.</p>
ThisPollThreshold	eth:100 IPoIB:100	<p>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.</p> <p>The valid values are 0 up to 1000.</p> <p>Note: This registry value is not exposed via the UI.</p>
*HeaderDataSplit	eth:0 IPoIB:0	<p>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.</p> <p>The valid values are:</p> <ul style="list-style-type: none"> • 0: disable • 1: enable <p>Note: This registry value is not exposed via the UI.</p>

Value Name	Default Value	Description
VlanId	eth:0 IPoIB:0	<p>Enables packets with VlanId. It is used when no team intermediate driver is used.</p> <p>The valid values are:</p> <ul style="list-style-type: none"> • 0: disable <p>No Vlan Id is passed.</p> <ul style="list-style-type: none"> • 1-4095 <p>Valid Vlan Id that will be passed.</p> <p>Note: This register cannot be configured via a VF that is running over Microsoft Hyper-V.</p>
TxForwardingProcessor	Automatically selected based on RSS configuration	<p>The processor that will be used to forward the packets sent by the forwarding thread.</p> <p>Default is based on number of rings and number of cores on the machine.</p> <p>Note: This registry value is not exposed via the UI.</p>
DefaultRecvRingProcessor	Automatically selected based on RSS configuration	<p>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.</p> <p>Note: This registry value is not exposed via the UI.</p>
TxInterruptProcessor	Automatically selected based on RSS configuration	<p>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.</p> <p>Note: This registry value is not exposed via the UI.</p>
*NumRSSQueues	eth:8 IPoIB:8	<p>The maximum number of the RSS queues that the device should use.</p> <p>Note: The maximum value of this registry key is 64.</p> <p>Note: This registry key is only in Windows Server 2012 and above.</p>

Value Name	Default Value	Description
BlueFlame	eth:1 IPoIB:1	<p>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).</p> <p>The valid values are:</p> <ul style="list-style-type: none"> • 0: disable • 1: enable <p>Note: This registry value is not exposed via the UI.</p>
*MaxRSSProcessors	eth:8 IPoIB:8	<p>The maximum number of RSS processors.</p> <p>Note: This registry key is only in Windows Server 2012 and above.</p>

4.6.6 Ethernet Registry Keys

The following section describes the registry keys that are only relevant to Ethernet driver.

Value Name	Default Value	Description
RoceMaxFrameSize	1024	<p>The maximum size of a frame (or a packet) that can be sent by the RoCE protocol (a.k.a Maximum Transmission Unit (MTU)).</p> <p>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</p> <p>The valid values are:</p> <ul style="list-style-type: none"> • 256 • 512 • 1024 • 2048 <p>Note: This registry key is supported only in Ethernet drivers.</p>
*PriorityVLANTag	3 (Packet Priority & VLAN Enabled)	<p>Enables sending and receiving IEEE 802.3ac tagged frames, which include:</p> <ul style="list-style-type: none"> • 802.1p QoS (Quality of Service) tags for priority-tagged packets. • 802.1Q tags for VLANs. <p>When this feature is enabled, the Mellanox driver supports sending and receiving a packet with VLAN and QoS tag.</p> <p>Note: This register cannot be configured via a VF that is running over Microsoft Hyper-V.</p>

Value Name	Default Value	Description
PromiscuousVlan	0	<p>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.</p> <p>The valid values are:</p> <ul style="list-style-type: none"> • 0: disable • 1: enable <p>Note: This registry value is not exposed via the UI.</p>
UseRSSForRawIP	1	<p>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.</p> <p>The valid values are:</p> <ul style="list-style-type: none"> • 0: disable • 1: enable <p>This is also relevant for IPoIB.</p> <p>Note: This registry value is not exposed via the UI.</p> <p>Note: This registry key is applicable to the Physical Function (PF) only.</p>
UseRSSForUDP	1	<p>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.</p> <p>The valid values are:</p> <ul style="list-style-type: none"> • 0:disabled • 1: Enabled <p>Note: This registry value is not exposed via UI.</p> <p>Note: This registry key is applicable to the Physical Function (PF) only.</p>
SingleStream	0	<p>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</p> <p>The valid values are:</p> <ul style="list-style-type: none"> • 0:disabled • 1: Enabled <p>Note: Only relevant for Ethernet and IPoIB</p>

Value Name	Default Value	Description
IgnoreFCS	0	<p>The valid values are:</p> <ul style="list-style-type: none"> • 0: disabled • 1: enabled <p>When enabled, the device is configured to:</p> <ol style="list-style-type: none"> 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).

4.6.6.1 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.

Value Name	Default Value	Description
PerPriRxPause	0	<p>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.</p> <p>Notes:</p> <ul style="list-style-type: none"> • 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).
PerPriTxPause	0	<p>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.</p> <p>Notes:</p> <ul style="list-style-type: none"> • 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).

4.6.6.2 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](http://msdn.microsoft.com/en-us/library/windows/hardware/ff571034(v=vs.85).aspx)

Value Name	Default Value	Description
*RssOrVmqPreference	0	<p>Specifies whether VMQ capabilities should be enabled instead of receive-side scaling (RSS) capabilities.</p> <p>The valid values are:</p> <ul style="list-style-type: none"> • 0: Report RSS capabilities • 1: Report VMQ capabilities <p>Note: This registry value is not exposed via the UI.</p>
*VMQLookaheadSplit	1	<p>Specifies whether the driver enables or disables the ability to split the receive buffers into lookahead and post-lookahead buffers.</p> <p>The valid values are:</p> <ul style="list-style-type: none"> • 0: disable • 1: enable
*VMQVlanFiltering	1	<p>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.</p> <p>The valid values are:</p> <ul style="list-style-type: none"> • 0: disable • 1: enable
MaxNumVmq	127	<p>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.</p> <p>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.</p> <p>The valid values are 1 up to 127.</p> <p>Note: This registry value is not exposed via the UI.</p>
MaxNumMacAddrFilters	127	<p>The number of different MAC addresses that the physical port supports.</p> <p>This registry key affects the number of supported MAC addresses that is reported to the OS.</p> <p>The valid values are 1 up to 127.</p> <p>Note: This registry value is not exposed via the UI.</p>
MaxNumVlanFilters	125	<p>The number of VLANs that are supported for each port. The valid values are 1 up to 127.</p> <p>Note: This registry value is not exposed via the UI.</p>

4.6.7 IPoB Registry Keys

The following section describes the registry keys that are unique to IPoB.

Value Name	Default Value	Description
GUIDMask	0xE7	<p>Controls the way the MAC is generated for IPoIB interface. The driver uses the 8 bytes GUID to generate 6 bytes MAC.</p> <p>This value should be either 0 or contain exactly 6 non-zero digits, using binary representation.</p> <p>Zero (0) mask indicates its default value: 0xb' 11100111. That is, to take all, except intermediate bytes of GUID to form the MAC address.</p> <p>In case of an improper mask, the driver uses the default one.</p> <p>Note: This registry value is not exposed via the UI.</p>
MediumType802_3	0	<p>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.</p> <p>The valid values are:</p> <ul style="list-style-type: none"> • 0 - the interface is exposed as NdisPhysicalMediumInfiniband • 1 - the interface is exposed as NdisPhysicalMedium802_3. <p>Note: This registry value is not exposed via the UI.</p>
SaTimeout	1000	<p>The time, in milliseconds, before retransmitting an SA query request.</p> <p>The valid values are 250 up to 60000.</p>
SaRetries	10	<p>The number of times to retry an SA query request. The valid values are 1 up to 64.</p>
McastIcmpMldGeneralQueryInterval	3	<p>The number of runs of the multicast monitor before a general query is initiated. This monitor runs every 30 seconds.</p> <p>The valid values are 1 up to 10.</p>
LocalEndpointMaxAge	5	<p>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.</p> <p>The valid values are 1 up to 20.</p> <p>Note: This registry value is not exposed via the UI.</p>
LocalEndpointMonitorInterval	60000	<p>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.</p> <p>The valid values are 10000 up to 1200000.</p> <p>Note: This registry value is not exposed via the UI.</p>

Value Name	Default Value	Description
EnableQPR	0	Enables query path record. The valid values are: <ul style="list-style-type: none"> 0 - disable 1 - enable
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.

4.6.8 General Registry Values

This section provides information on general registry keys that affect Mellanox driver operation.

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: <ul style="list-style-type: none"> 0: Don't check 1: Check
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.

4.6.9 MLX BUS Registry Keys

4.6.9.1 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 [Finding the Index Value of the HCA](#).

Key Name	Key Type	Values	Description
PermitSriov	REG_DWORD	0 = Disabled 1 = Enabled (default)	Configures the SR-IOV mode.
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 SetNetAdapterSriov 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 / (MaxVFPort1+MaxVFPort2))*number of VFs burnt in firmware.

4.6.9.2 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.

Parameters	Parameter Type	Description	Allowed Values and Default
roce_mode	DWORD	Sets the RoCE mode. The following are the possible RoCE modes: <ul style="list-style-type: none"> • RoCE MAC Based (v1) • RoCE IP Based (v1) • RoCE over UDP (v2) • No RoCE 	<ul style="list-style-type: none"> • RoCE MAC Based = 0 • RoCE IP Based = 5 • RoCE over UDP = 2 • No RoCE = 4 <p>Default: No RoCE</p> <p>Default: RoCE over UDP = 2</p> <p>Note: The default value depends on the WinOF package used.</p>
roce_udp_port	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.	<ul style="list-style-type: none"> • 1 - 65535 <p>Default (IANA Port): 4791</p>

4.6.9.3 General Registry Keys

Registry Keys location for machine configuration:

HKLM\SYSTEM\CurrentControlSet\Services\mlx4_bus\Parameters

Key Name	Key Type	Values	Description
UpdateGIDTimerFrequency	DWORD	0-10000 Default: 3000	Polling interval in milliseconds of local IP-address changes for updating RDMA IP-based GIDs.
RoceRoE2EFlowControl Enable	DWORD	Enabled: 1 Disabled: 0 Default: 1	Determines whether End-to-End Flow Control for RC connections is enabled.

4.7 Dump Me Now (DMN)

DMN is a bus driver (mlx4_bus.sys) feature that generates dumps and traces from various components, including hardware, firmware and software, upon internally detected issues (by the resiliency sensors), user requests (mlxtool). DMN is unsupported on VFs.

DMN dumps are crucial for offline debugging. Once an issue is hit, the dumps can provide useful information about the NIC's state at the time of the failure. This includes hardware state dumps, firmware traces and various driver component state and resource dumps.

For information on the relevant registry keys for this feature, please refer to [Configuration](#) below.

4.7.1 DMN Triggers and APIs

The DMN feature supports the following triggering APIs:

1. mlxtool.exe tool can be used to trigger DMN by running the dump-me-now debug subcommand:

```
mlxtool.exe dbg dump-me-now bus dev func
```

For example:

```
> mlxtool.exe dbg dump-me-now 8 0 0
```

The BDFs (bus device function) of the installed Mellanox devices can be found, using the command:

```
> mlxtool.exe show devices
```

2. An internal API between different driver components, in order to support generating DMN upon self-detected errors and failures (by the resiliency feature).

4.7.2 Dumps and Incident Folders

The DMN feature generates a directory per incident, where it places all of the needed NIC dump files.

The DMN incident directory name includes a timestamp, dump type, DMN event source and reason. It uses the following directory naming scheme:

dmn-<device name>-<type of DMN>-<source of DMN trigger>-<reason>--<timestamp>

Example:

dmn-GENERAL-SH-NA-4.13.2017-07.49.02.747

In this example:

1. The dump type is "general".
2. The DMN was triggered internally by the self-healing feature.
3. In this version of the driver the cause for the dump is not available in case of a self-healing trigger.
4. The dump was created on April 13th, 2017 at 747 milliseconds after 7:49:02 AM.

In this version of the driver, the DMN generates the following dump files upon a DMN event. Additional files will be added in the future:

1. MST dump - Adapter's configuration space registers contents.
2. Firmware commands dump - history of recent firmware commands and their status.
3. EQ dump - Commands event queue contents.
4. Firmware traces - Traces generated by the firmware, and collected by the driver
5. IOV objects state in case of SR-IOV-Setup



In this version of the driver, the firmware traces that are logged into the driver's WPP session are not an actual part of the DMN dump, and should be collected separately by the user.

DMN incident dumps are created under the DMN root directory, which can be controlled via the registry. The default is `\Systemroot\temp\Mlx4_Dump_Me_Now`.

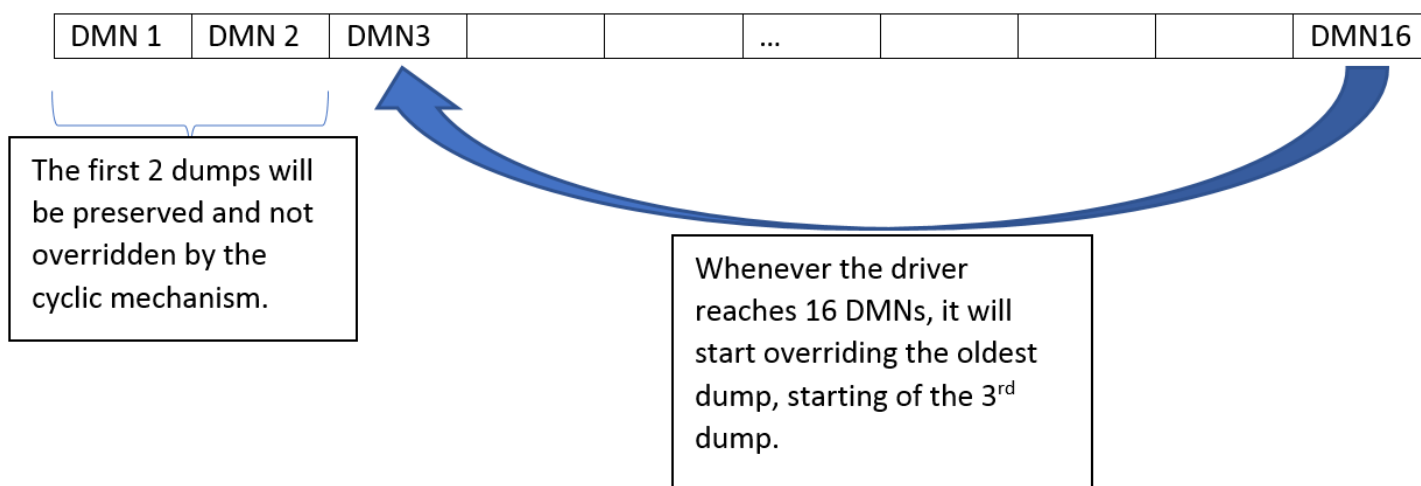
4.7.3 Cyclic DMN Mechanism

The driver manages the DMN incident dumps in a cyclic fashion, in order to limit the amount of disk space used for saving DMN dumps, and avoid low disk space conditions that can be caused from creating the dumps.

Rather than using a simple cyclic override scheme by replacing the oldest DMN incident folder every time it generates a new one, the driver allows the user to determine whether the first N incident folders should be preserved or not. This means that the driver will maintain a cyclic overriding scheme starting from a given index.

The two registry keys used to control this behavior are `DumpMeNowTotalCount`, which specifies the maximum number of allowed dumps under the DMN root folder, and `DumpMeNowPreservedCountMin`, which specifies the number of reserved incident folders that will not be overridden by the cyclic algorithm.

The following diagram illustrates the cyclic scheme's work, assuming `DumpMeNowPreservedCountMin=2` and `DumpMeNowTotalCount=16`:



4.7.4 Configuration

The registry keys for the DMN feature are located in: `HKLM\SYSTEM\CurrentControlSet\Services\mlx4_bus\Parameters`

The DMN dump is controlled by the following registry keys:

Dump Me Now Configurations

Key Name	Key Type	Default	Values	Description
DumpMeNowDirectory	REG_SZ	\Systemroot\temp\Mlx4_Dump_Me_Now	File system path	Path to the root directory in which the DMN places its dumps. The path should be provided in kernel path style, which means prefixing the drive name with "\\?\" (e.g. \\?\C:\DMN_DIR).
DumpMeNowTotalCount	REG_SZ	0-0xFFFF	128	The maximum number of allowed DMN dumps. Newer dumps beyond this number will override old ones.

Key Name	Key Type	Default	Values	Description
DumpMeNowPreservedCountMin	REG_SZ	0-0xFFFF	8	The number of DMN dumps that will be reserved and will never be overridden by newer DMN dumps.

Certain per-device resiliency registry keys determine when to trigger the DMN from the resiliency feature, and what sensors are allowed to perform that.

4.7.4.1 DMN-IOV Configuration

The DMN-IOV detail level can be configured by the "DmnIovMode" value that is located in device parameters registry key.

The default value is 2. The acceptable values are 0-4:

DMN-IOV Configuration

Values	Description
0	The feature is disabled
1	Major IOV objects and their state will be listed
2	All VF hardware resources and their state will be listed in the dump (QPs, CQs, MTTs, etc.)
3	All QP to Ring mapping will be added (the huge dump)
4	All IOV objects and their state will be list

4.7.5 Event Logs

Upon the success or failure of generating a dump, the DMN generates an event to the system event log. The following two events are used for that purpose:

Event Logs

Event ID	Severity	Message
0x100	Info	<device name>: The dump was created at folder (DMN folder name), due to dump-me-now request. Dump-me-now dumps are placed by default in folder %SystemRoot%\temp\Mlx4_Dump_Me_Now or a folder that was set by the registry keyword HKLM\SYSTEM\CurrentControlSet\Services\mlx4_bus\Parameters\DumpMeNowDirectory.
0x101	Error	<device name>: Failed to create a full dump me now. Dump me now root directory: <path to root DMN folder> Failure: <Failure description> Status: <status code>


4.8 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.

4.8.1 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](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](https://msdn.microsoft.com/library/cc853440(v=vs.85).aspx)

4.8.2 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.

4.8.2.1 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 the [RoCE Options](#) table.
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

Examples:

Linux server:

```
rping -v -s -a <IP address> -C 10
```

Windows client:

```
nd_rping -v -c -a <same IP as above> -C 10
```

4.9 Performance Tuning and Counters

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

For further information on WinOF performance, please refer to the *Performance Tuning Guide for Mellanox Network Adapters*.

⚠ 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.

This section contains:

- [General Performance Optimization and Tuning](#)
- [Application Specific Optimization and Tuning](#)
- [Tunable Performance Parameters](#)
- [Adapter Proprietary Performance Counters](#)
 - [Proprietary Mellanox Adapter Diagnostics Counters](#)
 - [Proprietary Mellanox QoS Counters](#)
 - [RSS Monitoring](#)
- [Device Proprietary Counters](#)

4.9.1 General Performance Optimization and Tuning

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

4.9.1.1 Mellanox Specific Extensions to the ND Interface

4.9.1.1.1 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

4.9.1.1.2 Nd2AdapterControlSetCqInterruptModeration

The method is an extension to the second version of the Network Direct SPI. It controls the amount of events received per completion, which reduces the amount of interrupts, and thereby improves the performance.

The method allows the user to control the amount of completions that will trigger an event, and the amount of time required before the next completion can occur, until an event is sent, as long as the completion number limit has not been reached.

The API for this interface is documented in the `ndspi_ext_mlx.h` SDK header file.

Using Nd2AdapterControlSetCqInterruptModeration

The usage of the Nd2AdapterControlSetCqInterruptModeration is similar to the usage of the function `NDK_FN_CONTROL_CQ_INTERRUPT_MODERATION` in MSDN NDK SPI. For more information, see: [https://msdn.microsoft.com/en-us/library/windows/hardware/jj552973\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/hardware/jj552973(v=vs.85).aspx)

The ModerationInterval will always be rounded down to its limit, thus the ModerationCount will never solely control the interrupt moderation on the CQ.

4.9.1.2 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.
```

4.9.1.3 Enable RSS

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

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

4.9.1.4 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 [Registry Tuning](#) above, or can be set post-installation manually.

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

1. Start the "Device Manager" (open a command line window and enter: devmgmt.msc).
2. Open "Network Adapters".
3. Select Mellanox IPoIB adapter, right click and select Properties.
4. Select the “Performance tab”.
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.
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\PerformanceTuning.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).

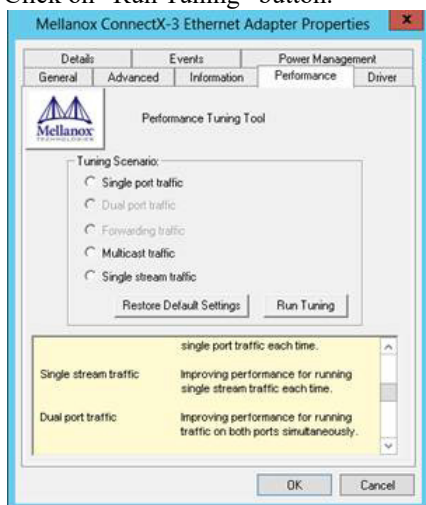
⚠ A reboot may be required for the changes to take effect.

4.9.1.5 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 [Registry Tuning](#) above. Specific scenarios tuning can be set post-installation manually.

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

1. Start the "Device Manager" (open a command line window and enter: devmgmt.msc).
2. Open "Network Adapters".
3. Select Mellanox Ethernet adapter, right click and select Properties.
4. Select the "Performance tab".
5. 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.



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\PerformanceTuning.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.

4.9.1.6 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>]
```

Performance Tuning Tool Application Options

Flag	Description
-s	<p>Single port traffic scenario.</p> <p>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.</p> <p>This option automatically sets:</p> <ul style="list-style-type: none"> • SendCompletionMethod = 0 • RecvCompletionMethod = 2 • *ReceiveBuffers = 1024 • In Operating Systems support NDIS6.3: RssProfile = 4 <p>Additionally, this option chooses the best processors to assign to:</p> <ul style="list-style-type: none"> • DefaultRecvRingProcessor • TxInterruptProcessor • TxForwardingProcessor • In Operating Systems support NDIS6.2: RssBaseProcNumber MaxRssProcessors • In Operating Systems support NDIS6.3: NumRSSQueues RssMaxProcNumber

Flag	Description
-d	<p>Dual port traffic scenario. This option must be followed by two connection names. The tuning in this case is codependent. This option automatically sets:</p> <ul style="list-style-type: none"> • SendCompletionMethod = 0 • RecvCompletionMethod = 2 • *ReceiveBuffers = 1024 • In Operating Systems support NDIS6.3: RssProfile = 4 <p>Additionally, this option chooses the best processors to assign to:</p> <ul style="list-style-type: none"> • DefaultRecvRingProcessor • TxForwardingProcessor • In Operating Systems support NDIS6.2: RssBaseProcNumber MaxRssProcessors • In Operating Systems support NDIS6.3: NumRSSQueues RssMaxProcNumber
-f	<p>Forwarding traffic scenario. This option must be followed by two connection names. The tuning in this case is codependent. This option automatically sets:</p> <ul style="list-style-type: none"> • SendCompletionMethod = 1 • RecvCompletionMethod = 0 • *ReceiveBuffers = 4096 • UseRSSForRawIP = 0 • UseRSSForUDP = 0 <p>Additionally, this option chooses the best processors to assign to:</p> <ul style="list-style-type: none"> • DefaultRecvRingProcessor • TxInterruptProcessor • TxForwardingProcessor • In Operating Systems support NDIS6.2: RssBaseProcNumber MaxRssProcessors • In Operating Systems support NDIS6.3: NumRSSQueues RssMaxProcNumber
-m	<p>Manual configuration This option must be followed by one connection name. This option assigns the provided base and number of CPUs to:</p> <ul style="list-style-type: none"> • *RssBaseProcNumber • *MaxRssProcessors <p>Additionally, this option assigns the following with processors inside the range:</p> <ul style="list-style-type: none"> • DefaultRecvRingProcessor • TxInterruptProcessor

Flag	Description
-r	<p>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:</p> <ul style="list-style-type: none"> • 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	<p>Specifies base RSS processor number. See examples. Used for manual option (-m) only.</p>
-n	<p>Specifies number of RSS processors. See examples. Used for manual option (-m) only.</p>
-st	<p>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:</p> <ul style="list-style-type: none"> • SendCompletionMethod = 0 • RecvCompletionMethod = 2 • *ReceiveBuffers = 1024 • In Operating Systems support NDIS6.3: RssProfile = 4 <p>Additionally, this option chooses the best processors to assign to:</p> <ul style="list-style-type: none"> • 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"
```

4.9.1.7 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
```

4.9.1.8 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\RequireSecuritySignature
```

4.9.2 Application Specific Optimization and Tuning

4.9.2.1 Ethernet Performance Tuning

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

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

1. Start the "Device Manager" (open a command line window and enter: devmgmt.msc).
2. Open "Network Adapters".
3. Right click the relevant Ethernet adapter and select Properties.
4. Select the "Advanced" tab
5. Modify performance parameters (properties) as desired.

4.9.2.1.1 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.

4.9.2.1.2 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:

1. Start the "Device Manager" (open a command line window and enter: devmgmt.msc).
2. Open "Network Adapters".
3. Right click the relevant IPoIB adapter and select Properties.
4. Select the "Advanced" tab.
5. Modify performance parameters (properties) as desired.


4.9.3 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
- **Send Buffers:** The number of send buffers
- **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).

4.9.4 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

4.9.4.1 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.

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.
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 sub-set 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.

Note 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 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.

4.9.4.2 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.

Mellanox Adapter Diagnostics Counter	Description
Requester length errors	Number of local length errors when the local machine generates outbound traffic.

Mellanox Adapter Diagnostics Counter	Description
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	<p>Number of local CQE with errors when the local machine receives inbound traffic.</p> <p>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.</p> <p>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”.</p>
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.
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.

Mellanox Adapter Diagnostics Counter	Description
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).
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).

Mellanox Adapter Diagnostics Counter	Description
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.
TX Copied Packets	Number of packets copied internally after either exceeding the size of the fragment list, or not reaching the minimum packet size.

4.9.4.3 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.

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}).
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}).

Mellanox QoS Counters	Description
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.

4.9.4.4 RSS Monitoring

The RSS monitoring feature allows users to track the packets received by the Mellanox adapters on their machine. The feature is comprised of two parts:

- Non-RSS Packet Capturing
 - A command-line interface for capturing received non-RSS packets.
- Packet monitoring through the performance monitor
 - A counter in the performance monitor for tracking the received non-RSS packets

Both parts of the feature must be enabled (per adapter instance), using the registry key: "RssMonitoringEnabled". Restart the adapter for these changes to take effect.

Registry key location:

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

For instructions on how to identify the interface index of a network interface in the registry <nn>, please refer to [Finding the Index Value of the Network Interface](#).

Controlling the RSS Monitoring Registry Key Feature

Key Name	Key Type	Values	Description
RssMonitoringEnabled	REG_DWORD	[0, 1] Default - Registry Key is not created (0)	Disable and enable the RSS Monitoring Feature (including packet capturing and tracking through the performance monitor) Value 0 - Disable the feature Value 1 - Enable the feature

4.9.4.4.1 Non-RSS Packet Capturing

This feature is added as a flag to "mlxtool.exe", which is installed as part of the WinOF package. The tool can be used to capture the Non-RSS packets received by the machine, and dump the captured traffic into a PCAP file.

Tool Default Path:

C:\Program Files\Mellanox\MLNX_VPI\Tools\mlxtool.exe

4.9.4.4.1.1 Using the Tool

Step 1. Enable the feature through the registry, as described in [RSS Monitoring](#) above.

Step 2. Obtain the name of the Mellanox adapter on your machine.

```
> mlxtool.exe show ports
```

Step 3. Query the status of the packet capturing. This can be used to tell if the tool is currently running:

```
> mlxtool.exe dbg nonrss-capture "Mellanox Adapter Name" query
```

Step 4. Start the packet capturing:

- When activating the feature, the user can specify two parameters:
 - Circular Buffer Size>: The number of packets to store in the capture buffer. This buffer is flushed when "collect" is called (see #5 for an description of collect). This is a circular buffer, so once it is full, it will override previous entries. If not specified, the default will be used. Default: 1024 packets.
 - <Number of Bytes to Capture per Packet>: The number of bytes to capture from each packet. Users can choose to only capture a "snip" of each packet. If not specified, the default will be used. Default: 128 bytes.
- In case of VPort mode (when NicSwitch is enabled), these parameters are global for all VPorts. The feature is either enabled on all VPorts or disabled on all VPorts. The circular buffer size and number of bytes to capture per packet cannot be specified per VPort.

```
> mlxtool.exe dbg nonrss-capture "Mellanox Adapter Name" start <Circular Buffer Size> <Number of Bytes to Capture per Packet>
```

Step 5. Collect the captured packets:

- Packets are stored in a circular buffer. They are dumped to a PCAP file when "collect" is called. This must be called before calling "stop", since the buffer gets cleared once the tool is stopped. The status of the feature can be retrieved using the "query" command described in #3.
- When capturing the dump, the user can specify two parameters:
 - <PCAP file name>: The name of the packet dump file. If not specified, the default will be used. Default: nonrss.pcap.
 - <VPort ID>: The Vport ID to collect for. This is only valid when the driver is operating in VPort mode (NicSwitch is enabled). If not specified, the default will be used. Default: 0.

```
> mlxtool.exe dbg nonrss-capture "Mellanox Adapter Name" collect <PCAP File Name> <VPort ID>
```

Step 6. Stop the packet capture.

- If "collect" is not called before the packet capture is stopped, the buffer that contains the captured packets will be cleared. Call "collect" before stopping the feature to dump the packets to a PCAP file.

```
> mlxtool.exe dbg nonrss-capture "Mellanox Adapter Name" stop
```

4.9.4.4.1.2 Packet Monitoring through the Performance Monitor

This feature activates counters in the performance monitor which can be used to monitor the packets that are being received by the Mellanox adapter(s). The feature allows the user to monitor incoming traffic for each adapter. It is split up per VPort (if NicSwitch is enabled) and per CPU.

All available counters are listed in the following figure:

Mellanox Adapter Rss Counters

- Encapsulated NonRss IPv4 Only
- Encapsulated NonRss IPv4/Tcp
- Encapsulated NonRss IPv4/Udp
- Encapsulated NonRss IPv6 Only
- Encapsulated NonRss IPv6/Tcp
- Encapsulated NonRss IPv6/Udp
- Encapsulated NonRss Misc
- Encapsulated Rss IPv4 Only
- Encapsulated Rss IPv4/Tcp
- Encapsulated Rss IPv4/Udp
- Encapsulated Rss IPv6 Only
- Encapsulated Rss IPv6/Tcp
- Encapsulated Rss IPv6/Udp
- Encapsulated Rss Misc
- NonRss IPv4 Only
- NonRss IPv4/Tcp
- NonRss IPv4/Udp
- NonRss IPv6 Only
- NonRss IPv6/Tcp
- NonRss IPv6/Udp
- NonRss Misc
- Rss IPv4 Only
- Rss IPv4/Tcp
- Rss IPv4/Udp
- Rss IPv6 Only
- Rss IPv6/Tcp
- Rss IPv6/Udp
- Rss Misc

Using the Tool

Step 1. Enable the feature through the registry, as described in [RSS Monitoring](#) above. In addition to the "RssMonitoringEnabled" registry key, this feature has an additional registry key "RssCountersActivatedAtStartup". This counter will begin counting received packets as soon as the adapter is connected. If this is not set, the counter will begin counting received packets as soon as the counter is added to the performance monitor. Restart the adapter for these changes to take effect.

Registry key location:

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

For instructions on how to identify the interface index of a network interface in the registry <nn>, please refer to [Finding the Index Value of the Network Interface](#).

Disable and Enable RSS Monitoring On Adapter Startup with the RssCountersActivatedAtStartup Registry Key

Key Name	Key Type	Values	Description
RssCountersActivatedAtStartup	REG_DWORD	[0, 1] Default - Registry Key is not created (0)	Choose when to begin counting received packets. Value 0 - Begin counting packets when the counter is added to the performance monitor Value 1 - Begin counting packets on adapter startup

Step 2. Open up the Windows performance monitor and add a new counter. The counter will be called "Mellanox Adapter Rss Counters".

Step 3. Select and add the desired instances of the "Mellanox Adapter Rss Counters".

- Note: There will be instances for each Mellanox Adapter that has the proper registry keys set (see introduction to Rss Monitoring for more information on registry keys).
- Note: In VPort mode (when NicSwitch is enabled), there will be a counter available per VPort for each CPU. There is also a "Total" for each VPort, which will report the total packets received by that VPort.
In native mode, there will be a counter available per CPU. There will also be a "Total" for each adapter, which will report the total packets received by that adapter.

Step 4. Switch the graph type in the performance monitor to "Report".

Step 5. Once the counters are added, the counters will get incremented based on the traffic that is received.

4.9.4.4.2 Proprietary 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.

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

Mellanox Adapter RSS Counters	Description
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
NonRss IPv6 Only	Number of Ipv6Only packets that have no RSS hash calculated by HW
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

Mellanox Adapter RSS Counters	Description
Rss Misc	Number of received packets that have RSS hash calculated with unknown RSS hash type

4.9.4.4.2.1 Controlling 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.

4.9.4.4.2.2 Proprietary 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.

⚠ This counters set is relevant only for ETH ports.

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 Activity Counters	Description
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.

Note a. These counters are only implemented in NDK and are **not** implemented in NDSPI.

4.9.5 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.

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.

Name	Description
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.

4.9.5.1 Mellanox Proprietary WinOF Bus Counters

This set of counters contains device's low-level counters used for debugging and behavior analysis.

Mellanox WinOF Bus Counters	Description
PCI Back-pressure/sec	Device core clocks without PCIe read/write credits.
No-WQE Drops/sec	The amount of packet drops due to no available receive buffers in the host.
Scatter Back-pressure/sec	Device core clocks where the 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	The number of receive WQE cache lookups resulted in a hit.
Receive WQE cache lookup/sec	The number of receive WQE cache lookups.
SQ Miss/sec	Transmit-queue/Requester-QP context cache miss.

Mellanox WinOF Bus Counters	Description
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.
Internal Processor0 Maximum Latency	The longest internal processor[0] process cycle in microSec.
Internal Processor1 Maximum Latency	The longest internal processor[1] process cycle in microSec.
Internal Processor2 Maximum Latency	The longest internal processor[2] process cycle in microSec.
Internal Processor3 Maximum Latency	The longest internal processor[3] process cycle in microSec.
Internal processor executed commands	The number of commands executed by the internal processor due to driver request via HCR command interface.
Last Retransmitted QP	The last QP that performed retransmission - RC QP only.
Current QPS in error state	The number of QPs in error state due to async error (e.g. retry exceeded) or due to CMD with errors (e.g. 2eer_qp cmd).
QP priority update flow events	The number of QP priority/SL update events.
Transmission engine hang events	The number of SX execution engine hang events.
Current QPS in limited state	The number of QPs that are in a limited state.
Total QPS in limited state	The total number of QPs that were in limited state.

Mellanox WinOF Bus Counters	Description
Maximum QPS in limited state	Maximum number of QPs that were in limited state at the same time
MPT entries used for QP	The number of Memory Protection Table (MPT) entries used for QPs.
MPT entries used for CQ	The number of Memory Protection Table (MPT) entries used for CQs.
MPT entries used for EQ	The number of Memory Protection Table (MPT) entries used for EQs.
MPT entries used for MR	The number of Memory Protection Table (MPT) entries used for MRs.
MTT entries used for QP	The number of Memory Translation Table (MTT) entries used for QPs.
MTT entries used for CQ	The number of Memory Translation Table (MTT) entries used for CQs.
MTT entries used for EQ	The number of Memory Translation Table (MTT) entries used for EQs.
MTT entries used for MR	The number of Memory Translation Table (MTT) entries used for MRs.
CPU MEM-pages (4K) mapped by TPT for QP	The total number of CPU memory pages (4K) mapped by TPT for QPs.
CPU MEM-pages (4K) mapped by TPT for CQ	The total number of CPU memory pages (4K) mapped by TPT for CQs.
CPU MEM-pages (4K) mapped by TPT for EQ	The total number of CPU memory pages (4K) mapped by TPT for EQs.
CPU MEM-pages (4K) mapped by TPT for MR	The total number of CPU memory pages (4K) mapped by TPT for MRs.
Arrived RDMA CNPs	The total number of received CNP packets for both ports.
Packets discarded due to invalid QP	The number of packets discarded due to an invalid QP.

4.10 Resiliency

4.10.1 Device Self-Healing

The Self-Healing feature allows the WinOF driver to recover from various error states. The feature is responsible for:

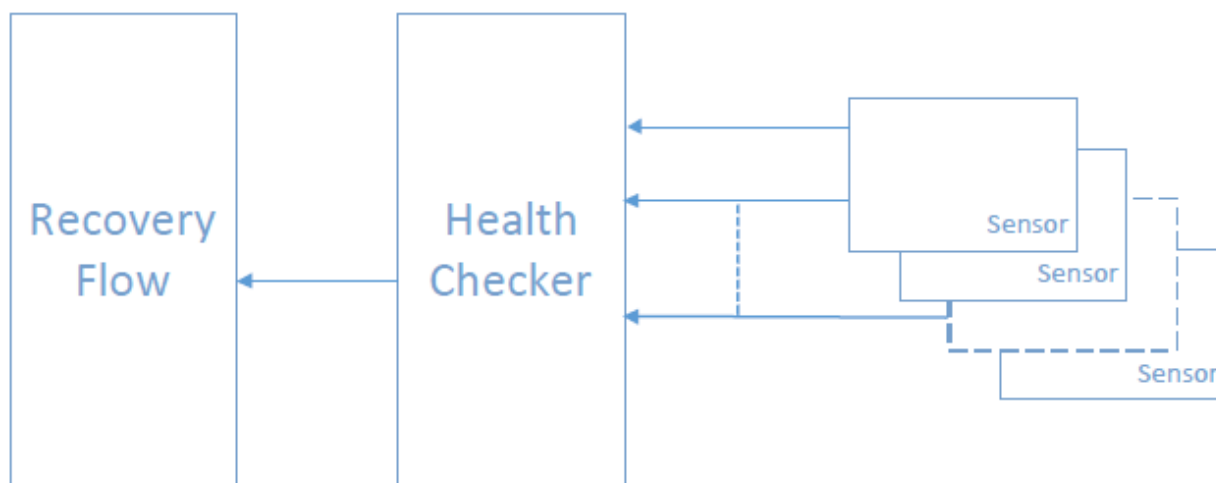
- Detecting of errors in the driver, firmware or hardware
- Performing the necessary actions for recovery
- Reporting the error and the action taken

The Self-Healing mechanism is comprised of two main components:

- Health-Checker: Determines when to trigger the recovery flow
- Recovery-Flows: Restarts the miniport driver. This component is comprised of:

- Miniport adapter restart
Recovery from miniport errors - in case the error is detected in the miniport driver, only the relevant miniport will be restarted

Self Healing



4.10.1.1 Health-Checker Mechanism

The driver's stacks run a continuous periodic loop of health-checking code that is designed to detect issues described in the "Sensors" section below.

Upon detecting an error, the health-checker reports the error to the self-healing manager, and it will determine whether any recovery steps should be performed, according to the configured policy.

The time interval for the periodic loop can be controlled by the user, per miniport driver instance, using the registry key: "SHCheckForHangTimeInSeconds".

Registry key location:

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

For instructions on how to find an interface index in the registry <nn>, please refer to [Finding the Index Value of the Network Interface](#).

Check-for-Hang SHCheckForHangTimeInSeconds Registry Key

Key Name	Key Type	Values	Description
SHCheckForHangTimeInSeconds	REG_DWORD	[1 - MAX_ULONG] Default: 4	The interval in seconds for the Check-for-Hang mechanism.

4.10.1.1.1 Sensors

The health state of the driver is examined by the designated sensors. Each sensor can be disabled/enabled independently. In case a specific sensor detects an error, it reports to the Self-Healing manager, logs an ETW message and executes the appropriate recovery-flow action.

In case the sensor is not activated, the error is ignored and no recovery flows are executed, but logs and dumps are generated.

4.10.1.1.1.1 Miniport Driver Sensors

The miniport sensors can be controlled per a miniport instance, using the per-miniport registry key paths.

Sensor	Description
Lack of Progress in Hardware for Ethernet Driver Send Queues	<p>The driver has posted a send WR in an Ethernet QP, but the hardware did not respond with a completion notification within a reasonable time period. The time period is determined according to the Check-for-Hang mechanism and "CheckForHangCQMaxNoProgress" registry key as the follows:</p> <ul style="list-style-type: none"> For each cycle where the Check-for-Hang identifies that the hardware does not respond with a completion notification, a dedicated counter for this test will be incremented. When the counter reaches the "CheckForHangCQMaxNoProgress" threshold, an error will be reported to the Self-Healing manager. The threshold for considering a posted WR as stuck is equal to $SHCheckForHangTimeInSeconds * CheckForHangCQMaxNoProgress$ seconds. In case the Head of Queue (HoQ) is disabled, this sensor will be ignored, and only dumps will be generated without performing any recovery steps. When the HoQ is disabled, posted WRs may complete after longer periods of time, due to congestion on the network, so this test is removed to avoid unnecessary recovery operations. <p>Mask: 0x00000008 ETW Event ID: 30009 Note: In VF, this sensor is always enabled and can not be disabled. This setting is defined in order to allow the Mellanox driver to go down at any time.</p>
Lack of Progress in Software for Ethernet Driver Receive Queues	<p>The driver has posted a receive WR to an Ethernet QP completed by the hardware, but it has not processed the completion event within a reasonable time period. The time is determined according to the same logic as described in "Lack of Progress in Hardware for Ethernet Driver Send Queues" row in this table.</p> <p>Mask: 0x00000040 ETW Event ID: 30011</p>
Receive Completion Error	<p>The hardware has reported an error in a receive WR.</p> <p>Mask: 0x00010000 ETW Event ID: 30021</p>
Send Completion Error	<p>The hardware has reported an error in a send WR.</p> <p>Mask: 0x00020000 ETW Event ID: 30022</p>
Lack of Progress in Software for Ethernet Driver Send Queues	<p>The driver has posted a send WR to an Ethernet QP completed by the hardware, but it has not processed the completion event within a reasonable time period. The time is determined according to the same logic as described in "Lack of Progress in Hardware for Ethernet Driver Send Queues" row in this table, but multiplied by two: $(2 * SHCheckForHangTimeInSeconds * CheckForHangCQMaxNoProgress)$.</p> <p>Mask: 0x00000010 ETW Event ID: 30029</p>

4.10.1.2 Configuration

The Self-Healing feature can be controlled by registry keys. The driver detects registry changes dynamically, and updates the Self-Healing settings automatically without requiring a driver restart.

4.10.1.2.1 Miniport Driver Registry Keys

Registry keys location:

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

For instructions on how to find an interface index in the registry <nn>, please refer to [Finding the Index Value of the Network Interface](#).

Miniport Driver Registry Keys

Key Name	Key Type	Values	Description
SHMPResetActiveSensorsMask	REG_DWORD	[0 , 0xFFFFFFFF] Default - 0xFFFFFFFF	Determines which sensors are active to execute a miniport reset upon error. For the sensors activation values, please refer to Miniport Driver Sensors above.
SHSensorsDumpMask	REG_DWORD	[0 , 0xFFFFFFFF] Default - 0x0	Determines which sensors are allowed to trigger the "Dump Me Now" feature upon error. For the sensors activation values, please refer to Miniport Driver Sensors above.
CheckForHangCQMaxNoProgress	REG_DWORD	[1 - 1000] Default: 4	The number of Check-for-Hang cycles with no progress in HW to count before reporting an error to the Self-Healing manager.

4.10.1.3 Logging

The self-healing manager records the following events in the System Event Viewer. Each record specifies the selected recovery flow and the reason to its execution:

Each sensor issues a unique ETW event upon error. The event could be found in the Windows event viewer, under "Applications and Services Log\Mellanox-Drivers\Operational". The following table contains all event messages:

Logging - Windows Event Viewer Applications Messages

Event ID	Message
3009	<Device name>: Lack of progress in hardware for Ethernet driver send queues sensor detected an error
30029	<Device name>: Lack of progress in software for Ethernet driver send queues sensor detected an error
30011	<Device name>: Lack of progress in software for Ethernet driver receive queues sensor detected an error
30021	<Device name>: Receive completion error sensor detected an error
30022	<Device name>: Send completion error sensor detected an error
30013	<Device name>: VF communication channel error sensor detected an error

The reasons are detailed in the following table:

Logging - Windows Event Viewer Messages

Event ID	Message
0x008b	<Device name>: Self Healing - Failed to activate the resiliency flow as a result of a SW reset failure, error=<error id>.%n The error was reported by the sensors <sensors id>.
0x008c	Restart <Interface name> as a result of an error that was reported by the sensors <Sensors mask> Self healing state: • Restarts count: <n>
0x008d	Stopped <Interface name> activity as a result of an error that was reported by sensors <n>.
0x0100	<Device name>: dump folder (<path>) was created due to a dump-me-now request.

4.11 RDMA Features

4.11.1 ND2 Provider Control

4.11.1.1 IND2ProviderControl Interface

4.11.1.1.1 Instantiating a NetworkDirect Provider Control

To use the new API call, the library's DllGetClassObject entry point must instantiate the IND2ProviderControl interface, using the IID_IND2ProviderControl iid.

Example:Disabling Mellanox Device while NetworkDirect Applications are Running

```
AutoRef<IClassFactory> aClassFactory;
//...
AutoRef<IND2ProviderControl> aProviderControl;
hr = aClassFactory->CreateInstance(NULL, IID_IND2ProviderControl, (void**)&aProviderControl);
RETURN_IF_FAILED(hr);
```

4.11.1.1.2 ND2RegisterForAdapterStatus

Once an IND2ProviderControl instance is instantiated, the following function can be called to register an adapter into the provider status mechanism:

Syntax:


```
HRESULT
Nd2RegisterForAdapterStatus(
__in IND2ProviderControl* pCtrl,
__in IND2_FN_ADAPTER_STATUS* pfnCallback,
__in void* pvContext,
__in_bcount(cbAddress) const struct sockaddr* pAddress,
__in ULONG cbAddress,
__deref_out void** ppHandle
)
```

Parameters:

The IND2ProviderControl instance is created by the IClassFactory::CreateInstance function.

- IND2ProviderControl [in]: The function callback that is called once a status change is detected for the specific adapter, detailed for the function parameters.
- pvContext [in]: A context value to associate with the registered adapter.
- pAddress [in]: A sockaddr buffer that specifies the local address to use for the shared endpoint. Typically, this is a sockaddr_in structure for IPv4 addresses and a sockaddr_in6 structure for IPv6 addresses.
If the sin_port or sin6_port member is specified as zero, the provider assigns a unique port number to the application from the ephemeral port range.
- cbAddress [in]: The size, in bytes, of the pAddress buffer.
- ppHandle [out]: The handle of the registered adapter. This parameter must be called with Nd2DeregisterForAdapterStatus before the final release of the Nd2RegisterForAdapterStatus instance.

Return value:

Return Code	Description
ND_SUCCESS	The operation succeeded.

4.11.1.1.3 IND2_FN_ADAPTER_STATUS

The following is the function callback signature that must be provided to the Nd2RegisterForAdapterStatus function to be called upon an adapter status change:

Syntax:

```
void CALLBACK IND2_FN_ADAPTER_STATUS(
__in void* pvContext,
__in_bcount(cbAddress) const struct sockaddr* pAddress,
__in ULONG cbAddress,
__in DWORD dwStatusFlags
);
```

Parameters:

- pvContext [in]: A context value to associate with the registered adapter.
- pAddress [in]: A sockaddr buffer that specifies the local address to use for the shared endpoint. Typically, this is a sockaddr_in structure for IPv4 addresses and a sockaddr_in6 structure for IPv6 addresses.
If the sin_port or sin6_port member is specified as zero, the provider assigns a unique port number to the application from the ephemeral port range.
- cbAddress [in]: The size, in bytes, of the pAddress buffer.
- dwStatusFlags [in]: The state flag of the new adapter status. Could be a single state flag or a combination of the following:

New Adapter Status State Flag Return Value:

Value	Meaning
0	The adapter is up and running
ND_ADAPTER_FLAG_ADAPTER_IS_DOWN	The adapter is down.
ND_ADAPTER_FLAG_ADAPTER_RDMA_DISABLED	RoCE mode is disabled In the adapter

4.11.1.1.4 Nd2DeregisterForAdapterStatus

The Nd2DeregisterForAdapterStatus function must be called to deregister a registered adapter from the IND2ProviderControl. Failing to call the function will result in unreleased allocated memory.

Syntax:

```
HRESULT
Nd2DeregisterForAdapterStatus(
    __in IND2ProviderControl* pCtrl,
    __in void* pHandle
)
```

Parameters:

- IND2ProviderControl [in]: The IND2ProviderControl created by the IClassFactory::CreateInstance function.
- pHandle [in]: The handle of the deregistered adapter.

Return value:

Return Code	Description
ND_SUCCESS	The operation succeeded.

Sample Code - Register:

```
static void CALLBACK
OnAdapterStatus(
__in void* pvContext,
__in_bcount(cbAddress) const struct sockaddr* pAddress,
__in ULONG cbAddress,
__in DWORD dwStatusFlags
){
    //.... //
    printf("flags = %x\n", dwStatusFlags); //Print the new adapter status
}

AutoRef<INetworkAddressInfo> aAddressInfo; // INetworkAddressInfo Object
//....
//CreateNetworkAddressInfo with aAddressInfo
//....
DWORD aContext; //Context for the specific adapter being registered
void* pvNd2RegisterForAdapterStatus = NULL; //AdapterStatus

hr = Nd2RegisterForAdapterStatus(
    aProviderControl,
    OnAdapterStatus,
    &aContext,
    aAddressInfo->GetAddr(),
    aAddressInfo->Length(),
    &pvNd2RegisterForAdapterStatus);
RETURN_IF_FAILED(hr);
//.....//
//.....//
    if (pvNd2RegisterForAdapterStatus) {
        hr = Nd2DeregisterForAdapterStatus(aProviderControl, pvNd2RegisterForAdapter-
Status);
    }
return hr;
```

4.11.2 Disabling Mellanox Device while NetworkDirect Applications are Running

Mellanox's NIC device can be disabled at any time. When the device is disabled while NetworkDirect (ND) applications are running, the following will occur:

- Any new request or operation that is executed by the ND application will fail explicitly
- All pending overlapped requests will be canceled
- All pending send requests will be completed with error (flush with error)
- All RDMA resources will be released
- The device will be marked as disabled

5 Utilities

This section describes the following utilities:

- [Snapshot Tool](#)
- [part_man - Virtual IPoIB Port Creation Utility](#)
- [InfiniBand Fabric Diagnostic Utilities](#)
- [Fabric Performance Utilities](#)
- [mlxtool](#)

5.1 Snapshot Tool

The snapshot tool scans the machine and provide information on the current settings of the operating system, networking and hardware.

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:

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.
2. Click on Generate HTML button:



Once the report is ready the folder which contains the report will be opened automatically.

5.2 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
```

Virtual IPoIB Port Creation Utility Options

Option	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.

! 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 vPoIB 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 vPoIB port, please refer to [Multiple Interfaces over Non-Default PKeys Support](#).

⚠ The `part_man` tool allows the creation of up to 64 vPoIB interfaces (32 per port).

5.3 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.

5.3.1 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 one 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 main page.

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 <code>ibportstate</code> can be used to <ul style="list-style-type: none"> •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.

Utility	Description
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.
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.

Utility	Description
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.
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.

5.4 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.

 For further information on the following tools, please refer to the help text of the tool by running the --help command line parameter.

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.

Utility	Description
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.
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.

5.5 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 5.50 package.

In order to extract the tool, please follow the steps in [Extracting Files Without Running Installation](#). Make sure to copy the tool and the following files to the same location: mlxtool.exe, complib.dll, mlxapi.dll.

5.5.1 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"
```

5.5.2 dbg sub-tool

This sub-tool is used to extract debug information.

Usage

```
mlxtool.exe dbg <sub-tool> <arguments>
```

5.5.2.1 nonrss-capture

This sub-tool is used to control the miniport driver's NonRss packet snip capturing. For more information, see Section 3.9.4.4.1, “Non-RSS Packet Capturing”, on page 159.

Usage

```
mlxtool dbg nonrss-capture <Interface Name>  
<"start"|"stop"|"query"|"collect"> [OTHER PARAMETERS]
```

Other Parameters

- For “start”, provide <Circular buffer size> <Number of bytes to capture per packet>
- For “collect”, provide <Output pcap file name> <VPortID in case of virtualization>

5.5.2.2 sw-reset

This sub-tool is used to trigger a software reset request.

Usage

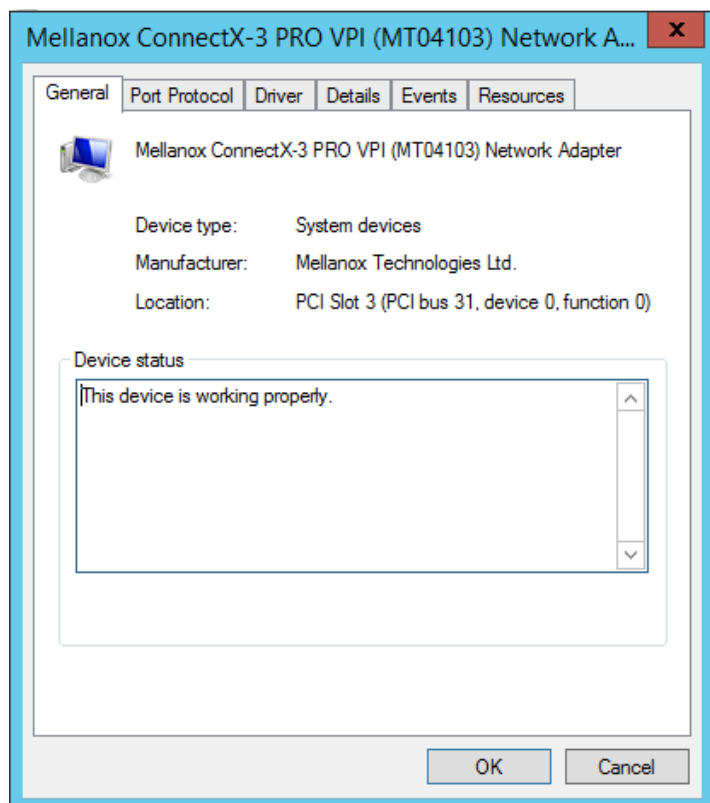
```
mlxtool dbg sw-reset <pci-bus#> <pci-device#> <pci-function#>
```

5.5.2.3 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 64, “Events Causing Automatic State Dumps,” on page 214.

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#>
```



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
```

5.5.2.4 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 4.6.3.4, “show port list”, on page 196.

5.5.2.5 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_IPC	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_RTR2RTR_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_RTS2RTR_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 4.6.2.3, “mstdump”, on page 189.

In order to view the list of all devices, please refer to Section 4.6.3.5, “show device list”, on page 196.

5.5.2.6 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

5.5.2.7 Resources

This tool is used for pulling resource tracker information for VFs. In addition, it displays the effective max (the max quotas supported by the machine for resource type minus the firmware quota reserved) for QPs, CQs, SRQs, MPTs, MTTs and FS_RULEs.

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, refer to Section 4.6.2.3, “mstdump”, on page 189.
In order to view the list of all devices, refer to Section 4.6.3.5, “show device list”, on page 196.

Usage:

```
mlxtool.exe dbg resources [<pci-bus#> <pci-device#> <pci-function#>]
```

Example:

If "Location" is "PCI Slot 3 (PCI bus 130, device 0, function 0)", type:

```
mlxtool.exe dbg resources 130 0 0
```

Device: 8:0.0		Total UFs: 3	

Resource:	max quota		
QPs:	983040		
CQs:	65536		
SRQs:	65536		
MPIs:	524288		
MTIs:	33554176		
FS_RULEs:	8192		

PF			
	Resource: count	Quota: count	Reserved: count
	-----	-----	-----
QPs:	4491	Quota: 32768	Reserved: 136
CQs:	25	Quota: 16384	Reserved: 128
SRQs:	12	Quota: 16384	Reserved: 64
XRCDs:	0	Quota: 0	Reserved: 4
MPIs:	1	Quota: 16384	Reserved: 256
MTIs:	207	Quota: 1048640	Reserved: 64
MACs:	5	Quota: 128	Reserved: 0
ULANs:	0	Quota: 128	Reserved: 0
EQs:	34	Quota: 128	Reserved: 28
COUNTERs:	20	Quota: 128	Reserved: 0
FS_RULEs:	18	Quota: Unlimited	Reserved: 0

UF #0	port #1		
	Resource: count	Quota: count	
	-----	-----	
QPs:	0	Quota: 1024	
CQs:	0	Quota: 512	
SRQs:	0	Quota: 512	
XRCDs:	0	Quota: 0	
MPIs:	0	Quota: 4096	
MTIs:	0	Quota: 16384	
MACs:	0	Quota: 1	
ULANs:	0	Quota: 0	
EQs:	0	Quota: 52	
COUNTERs:	0	Quota: 2	
FS_RULEs:	0	Quota: 1	

UF #1	port #1		
	Resource: count	Quota: count	
	-----	-----	
QPs:	194	Quota: 1024	
CQs:	5	Quota: 512	
SRQs:	3	Quota: 512	
XRCDs:	0	Quota: 0	
MPIs:	1	Quota: 4096	
MTIs:	15	Quota: 16384	
MACs:	1	Quota: 1	
ULANs:	0	Quota: 0	
EQs:	4	Quota: 52	
COUNTERs:	1	Quota: 2	
FS_RULEs:	1	Quota: 1	

UF #2	port #1		

If PCI is not provided, the information is shown for of all the devices.

5.5.2.8 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	0xffffffff	0xc000	0xFF12-401B-FFFF-00000000000000000001
333300000001	0xffffffff	0xc000	0xFF12-601B-FFFF-00000000000000000001
ffffffffffff	0xffffffff	0xc000	0xFF12-401B-FFFF-000000000000FFFFFFFF
3333ff9bd3f1	0xffffffff	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.

5.5.2.9 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 4.6.2.3, "mstdump", on page 189.

In order to view the list of all devices, please refer to Section 4.6.3.5, "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.

5.5.2.10 Restart

Restart interface network adapter.

Usage:

```
mlxtool dbg restart <Interface Name>
```

5.5.2.11 Port Diagnostic Database Register

The tool provides troubleshooting and operational information that can assist in debugging physical layer link related issues.

Usage:

Example:

```
mlxtool dbg pddrinfo 2 0 0 1
```

Please note, if the PCI is not specified, the PDDR information for each port will apply to all devices.

Example:

```
mlxtool dbg pddrinfo
```

Sample Output:

```
Bus:5 Device:0 Function:0
Operational Info for Port# 2:
-----
State                : Active
Physical State       : ETH_AN_FSM_AN_GOOD
Speed                : 40 Gbps
FEC                  : No FEC
Loopback Mode        : No Loopback
Supported speeds      : 0x21045 < CX-1G CR-10G CX4-10G CR4-40G CR4-56G  >

Bus:5 Device:0 Function:0

Troubleshoot Info Port# 2
-----
Status Opcode        : 0
Group Opcode         : 0
Message              : No issues detected
```

5.5.3 show

This tool is used to show specific information.

Usage:

```
mlxtool.exe show <tool-name> <tool-arguments>
```

5.5.3.1 show packet-filter

This tool is used to show the packet filter configuration.

Usage:

```
mlxtool show packet-filter [<Interface Name>]
```

Example:

```
mlxtool show packet-filter "Ethernet 5"
```

```
Ethernet port: Ethernet 5

Packet Filter configuration: 0xb
    NDIS_PACKET_TYPE_DIRECTED
    NDIS_PACKET_TYPE_MULTICAST
    NDIS_PACKET_TYPE_BROADCAST
```

5.5.3.2 show qos

This tool is used to show the QOS settings.

Usage:

```
mlxtool show qos [<pci-bus#> <pci-device#> <pci-function#>]
```

5.5.3.3 show nd

This tool is used to show the ND connections.

Usage:

```
mlxtool show nd connections [-a <IP address>] [-p <Process Id>] [-e <error>] [-n count][-t time]
```

where:

```
-a|--address <IP address>] - enumerate ND connections of this ND adapter.(default:all)
-p|--pid Process Id        - enumerate ND connections for the process with this Process Id.
(default:all)
-e|--error                  - Show QPs in error state
-n|--count                  - Number of QPs(in case of -e) or EPs to fetch at one go from
NDFLTR(default:1000,max:10000)
-t|--time                   - Time interval between each info fetch in seconds(Can only be run
with -e flag. Default: 2seconds)
```

5.5.3.4 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
```

5.5.3.5 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
```

5.5.3.6 Show vxlan

This tool is used to show VXLAN offload configuration.

Usage:

```
mlxtool.exe show vxlan [<Interface Name>]
```

Example:

If you wish to display the information of a specific Ethernet interface, for example "Ethernet 12", run the following command:

```
mlxtool.exe show vxlan "Ethernet 12"
```

```
Ethernet port: Ethernet 12
UxLAN Offload enabled: false
Uxlan_offload_params.udp_dport: 4789
```

If the interface name is not provided, the information is shown for of all the Ethernet interfaces.

5.5.3.7 Show selfhealing

This tool is used to show sensors state and resets count.

Usage:

```
mlxtool show selfhealing <tool-name> <tool-arguments>
```

5.5.3.7.1 show selfhealing port

This tool is used to show sensors state and resets count for an interface.

Usage:

```
mlxtool show selfhealing port [Interface Name]
```

If an interface name is not provided, the information is shown for of all the interfaces.

Example:

```
mlxtool show selfhealing port "Ethernet 7"
```

```
Ethernet NIC: Ethernet 7
-----

Sensors state to execute MP reset upon error:
-----
| Sensor | State |
|-----|-----|
| Lack of Progress in Hardware for Ethernet Driver Send Queues | OFF |
| Lack of Progress in Software for Ethernet Driver Send Queues | OFF |
| Lack of Progress in Software for Ethernet Driver Receive Queues | OFF |
| Receive Completion Error | OFF |
| Send Completion Error | OFF |
|-----|-----|

Sensors state to execute HCA reset in case the MP reached max resets count:
-----
| Sensor | State |
|-----|-----|
| Lack of Progress in Hardware for Ethernet Driver Send Queues | ON |
| Lack of Progress in Software for Ethernet Driver Send Queues | ON |
| Lack of Progress in Software for Ethernet Driver Receive Queues | ON |
| Receive Completion Error | ON |
| Send Completion Error | ON |
|-----|-----|

Sensors state to execute HCA reset upon error without executing MP reset:
-----
| Sensor | State |
|-----|-----|
| Lack of Progress in Hardware for Ethernet Driver Send Queues | ON |
| Lack of Progress in Software for Ethernet Driver Send Queues | ON |
| Lack of Progress in Software for Ethernet Driver Receive Queues | ON |
| Receive Completion Error | ON |
| Send Completion Error | ON |
|-----|-----|

The number of MP resets already executed: 0
The max number of MP resets that can be executed: 4096
```

5.5.3.8 regkeys

This tool is used to show registry keys information.

Usage:

```
mlxtool.exe show regkeys <tool-name> <tool-arguments>
```

5.5.3.8.1 show regkeys config

This tool is used to show registry keys configurations from the driver.

Usage:

```
mlxtool.exe show regkeys config [<Eth Interface Name>]
```

For the information of interface "Ethernet 5", type:

```
mlxtool.exe show regkeys config "Ethernet 5"
```

Sample Output:

```

ETH NIC: Ethernet 5
-----

Registry Key          Config
NumTcb                16
*RSS                  1
*MaxRssProcessors     8
*NumRSSQueues         8
*InterruptModeration  1
*IPChecksumOffloadIPv4 3
*TCPUDPChecksumOffloadIPv4 3
*TCPUDPChecksumOffloadIPv6 3
*LsoV2IPv4            1
*LsoV2IPv6            1
TxBwPrecedence        0
IgnoreFCS              0
*ReceiveBuffers       4096
*TransmitBuffers       2048
*JumboPacket          9870
NdkkWithGlobalPause   1
RoceMaxFrameSize       2048
*FlowControl          3
*PriorityVLANTag       3

```

If the interface name is not specified, the information is shown for of all the Ethernet interfaces.

5.5.3.8.2 Show regkeys diff

This tool used to show registry keys settings from registry and driver.

Usage:

```
mlxtool.exe show regkeys config [<Eth Interface Name>]
```

For the information of interface "Ethernet 5", type:

```
mlxtool.exe show regkey diff "Ethernet 5"
```

Sample Output:

```

ETH NIC: Ethernet 4
-----

Registry Key          Registry Config    Current
NumTcb                16                16
*RSS                  1                 1
*MaxRssProcessors     8                 8
*NumRSSQueues         8                 8
*InterruptModeration  1                 1
*IPChecksumOffloadIPv4 3                 3
*TCPUDPChecksumOffloadIPv4 3                 3
*TCPUDPChecksumOffloadIPv6 3                 3
*LsoV2IPv4            1                 1
*LsoV2IPv6            1                 1
TxBwPrecedence        0                 0
RxBufferAlignment     0                 0
IgnoreFCS              0                 0
*ReceiveBuffers       4096              4096
*TransmitBuffers       2048              2048
*JumboPacket          1514              1514
*PacketDirect         1                 1
NdkkWithGlobalPause   1                 1
*RssOnHostVPorts      1                 1

```

If the interface name is not specified, the information is shown for of all the Ethernet interfaces.

5.5.3.8.3 show regkeys info

This tool is used to show registry keys information: Default, Min and Max settings.

Usage:

```
mlxtool.exe show regkeys info [<Eth Interface Name>]
```

For the information of interface "Ethernet 5", type:

```
mlxtool.exe show regkeys info "Ethernet 5"
```

Sample Output:

Registry Key	Min	Max	Default
NumTcb	1	128	32
*RSS	0	1	1
*MaxRssProcessors	1	4294967295	8
*NumRSSQueues	1	4294967295	8
*InterruptModeration	0	1	1
*IPChecksumOffloadIPv4	0	3	3
*TCPUDPChecksumOffloadIPv4	0	3	3
*TCPUDPChecksumOffloadIPv6	0	3	3
*LsoV2IPv4	0	1	1
*LsoV2IPv6	0	1	1
TxBwPrecedence	0	1	0
RxBufferAlignment	0	63	0
IgnoreFCS	0	1	0
*ReceiveBuffers	256	8192	4096
*TransmitBuffers	256	4096	2048
*JumboPacket	614	9614	1514
*PacketDirect	0	1	0
NdkWithGlobalPause	0	1	0
*RssOnHostVPorts	0	1	1

If the interface name is not specified, the information is shown for of all the Ethernet interfaces.

5.5.3.9 show driverParams

The tool displays the default, min, max and current values of certain internal driver parameters which can be updated using the "modify" command to take effect, without requiring driver restart.

Usage:

```
mlxtool.exe show driverparams <interface name>
```

Example:

```
mlxtool.exe show driverparams "Ethernet 5"
```

Output Fields	Description
MaxNumberOfPacketsToIndicateDPC	Maximum number of RX packets to indicate to OS in Interrupt mode.
MaxNumberOfPacketsToIndicateThread	Maximum number of RX packets to indicate to thee OS in polling mode.
MaxCallsToNdisIndicate	Maximum number of times chained packets can be indicated before packets processing is stop processing is stopped.
MaxNumTxPacketsToIndicateComplete	Maximum number of TX packets to complete back to the OS in one call.
NicMinRfd	Minimum NIC receive frame descriptors to hold the incoming RX packets.

Output Fields	Description
UponMaxConsumedReceivesMoveToDPC	Move to Interrupt mode once the maximum RX packet processing limit is reached.
UponMaxConsumedTransmitsMoveToDPC	Move to Interrupt mode once we the maximum TX packet processing limit is reached.
TrackNBLReturnLatency	Enables the driver to calculate the time from when it indicated to the OS to the time it received the RX buffer back.
CountersModeFlags	Disable or enable caching statistics.
QueryStatsMsec	Timer in milliseconds to query statistics periodically.

Sample Output:

```
C:\Users\Administrator\Desktop>mlxtool.exe show driverparams "Ethernet 7"
```

Index	Driver Parameter	Current	Min	Max	Default
0	MaxNumberOfPacketsToIndicateDPC	128	10	2048	128
1	MaxNumberOfPacketsToIndicateThread	128	10	2048	128
2	MaxCallsToNdisIndicate	5	0	4096	5
3	MaxNumTxPacketsToIndicateComplete	16	1	128	16
4	NicMinRfd	1024	0	4096	1024
5	UponMaxConsumedReceivesMoveToDPC	0	0	1	0
6	UponMaxConsumedTransmitsMoveToDPC	0	0	1	0
7	TrackNBLReturnLatency	0	0	1	0
8	CountersModeFlags	0	0	1	4294967295
9	QueryStatMsec	200	100	2000	200

5.5.3.10 show perfstats

The tool displays runtime ring statistics (both RX and TX rings), and specific global statistics. The tool can be used to create snapshots, gather statistics for a certain time duration, or reset them.

Usage:

Example:

Additional Options:

- RX | TX: provides a snapshot of ring statistics
- RXD <time in seconds>|TXD <time in seconds>: provides ring statistics for a certain time duration
- RxReset|TxReset|GlobalReset: internally resets statistics
- ShowAll: dumps a snapshot of all TX, RX and global statistics
- ResetAll: internally resets all TX, RX and global statistics

Output Fields	Description
PacketsReceivedDPC	Number of packets processed in Interrupt mode.

Output Fields	Description
PacketsReceivedThread	Number of packets processed in Polling mode.
Average Packets Per DPC	Average number of packets indicated to OS in interrupt mode.
Average Packets Per Thread	Average number of packets indicated to OS in polling mode.
ConsumedMaxReceives	Number of times the max packet limit was processed (128 packets default)
NumTrafficProfileTransitions	Number of transfers from interrupt mode to polling mode
PacketsLowResources	Number of times packets with low resources were indicated. In this mode, the driver has immediate packet ownership.
SkipReceiveDueToControlChanges	Bail out from RX processing after entering it due to external factors.
DpcWatchDogSingleDpcStarvation	This counter is incremented each time DPC timer limit is hit (or nearing).
DpcWatchDogTotalDpcStarvation	This counter is incremented each time DPC watchdog limit is hit (or nearing).
Ring Throughput	Runtime Ring throughput in G/bps.
Histogram of Number Of Received Packets Per Indicate Call (non-low resources)	<p>Histogram displaying the number of times we indicated less than or equal to 'x' packets to the OS, in either DPC or polling mode.</p> <p>Example:</p> <p>Range Hits</p> <p>-----</p> <p>0- 39 4</p> <p>40 - 79 2</p>

Sample Output:

```
C:\Users\Administrator\Desktop>mlxtool.exe show perfstats "Ethernet 7" rx
**Per RX Queue statistics - Number of Total RX Queues is 4:
Parameters          | RxQ# 0          | RxQ# 1          | RxQ# 2          | RxQ# 3          |
CpuNumber            | 0               | 2               | 4               | 6               |
NumberOfInterrupts   | 150578          | 101247          | 46126           | 101119          |
PacketsReceivedDPC   | 5389658         | 2939551         | 1164346         | 2951631         |
PacketsReceivedThread | 0               | 0               | 0               | 0               |
AveragePkts/DPC      | 36              | 29              | 25              | 29              |
AveragePkts/Thread   | 0               | 0               | 0               | 0               |
ConsumedMaxReceives  | 0               | 0               | 0               | 0               |
NumTrafficProfileTransitions | 12              | 2               | 2               | 2               |
PacketsLowResources  | 0               | 0               | 0               | 0               |
SkipReceiveDueToControlChanges | 0              | 0               | 0               | 0               |
DpcWatchDogSingleDpcStarvation | 0              | 0               | 0               | 0               |
Ring Throughput in Gbps | 0.00           | 0.00           | 0.00           | 0.00           |
Average Pkt Size     | 0               | 0               | 0               | 0               |
rxReturnNblAvgLatency in us | -              | -              | -              | -              |
rxReturnNblMinLatency in us | -              | -              | -              | -              |
rxReturnNblMaxLatency in us | -              | -              | -              | -              |
Rx Pkts Per Indicate Call 0 - 39 | 153309         | 101247          | 46126           | 101119          |
Rx Pkts Per Indicate Call 40 - 79 | 0              | 0               | 0               | 0               |
Rx Pkts Per Indicate Call 80 - 119 | 0              | 0               | 0               | 0               |
Rx Pkts Per Indicate Call 120 - 128 | 0              | 0               | 0               | 0               |
Command executed successfully
```

5.5.4 modify Tool

This tool is used to modify driver parameters.

Usage:

```
mlxtool.exe modify <tool-name> <tool-arguments>
```

5.5.4.1 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).

6 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.

- [Installation Related Troubleshooting](#)
- [InfiniBand Related Troubleshooting](#)
- [Ethernet Related Troubleshooting](#)
- [Performance Related Troubleshooting](#)
- [Virtualization Related Troubleshooting](#)
- [Reported Driver Events](#)
- [Extracting WPP Traces](#)

6.1 RDMA Related Troubleshooting

Description	Troubleshooting
In case the network adapter of the remote ND connection was reset, the local peer does not receive a disconnect notification.	To make sure that the connection is still valid, post a send request on the QP, and check its completion status.

6.2 Installation Related Troubleshooting

6.2.1 Installation Error Codes and Troubleshooting

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: <ul style="list-style-type: none"> • 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: https://rcmtech.wordpress.com/2013/08/27/server-2012-remote-desktop-session-host-installation-hangs-at-windows-installer-coordinator/	Follow the recommendation in the article.

Issue	Cause	Solution
After driver upgrade, The ND state might be invalid, and the following event might appear in the event viewer: "Ndfldr: ND is in invalid state as a result of a mismatch between the ndfldr.sys driver version and mlx4_bus.sys driver version."	Upgrading the driver while the ND applications are active will hold the current ndfldr driver loaded, and prevent the loading of the updated driver.	Close all applications that use the ND, and restart the bus driver.

6.2.1.1 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>

6.2.1.2 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.

6.2.1.3 Restore Configuration Warnings

Error Code	Description	Troubleshooting
3	Failed to restore the configuration	Please see log for more details and contact the support team

6.3 InfiniBand Related Troubleshooting

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. Note: This issue can occur when using firmware version 2.40.5000. To avoid it, upgrade to version 2.40.5030 and above.

6.4 Ethernet Related Troubleshooting

For further performance related information, please refer to the *Performance Tuning Guide* and to [Performance Tuning and Counters](#).

Issue	Cause	Solution
Low performance.	Non-optimal system configuration might have occurred.	See section Performance Tuning and Counters 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.	<ol style="list-style-type: none"> 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". <p>or a less recommended suggestion (as it will cause low performance):</p> <ul style="list-style-type: none"> • Disable RSS on the adapter, run: "netsh int tcp set global rss = no dynamic balancing".

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, contact 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 Firmware Upgrade .
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.	<ul style="list-style-type: none"> Remove the value key under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\TcpWindowSize <p>Or</p> <ul style="list-style-type: none"> 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.

6.5 Performance Related Troubleshooting

Issue	Cause	Solution
Low performance issues	The OS profile might not be configured for maximum performance.	<ol style="list-style-type: none"> Go to "Power Options" in the "Control Panel". Make sure "Maximum Performance" is set as the power scheme 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 <ul style="list-style-type: none"> 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: <ul style="list-style-type: none"> 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: <ul style="list-style-type: none"> Set Global pause on the switches, routers. Run "Disable-NetAdapterQos *" on all of the hosts in a PowerShell window.

6.5.1 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: 56.0Gbps / 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:

- Run `"start /b /affinity 0x1 nd_send_bw -S <IP_host>"` where `<IP_host>` is the local IP.
- Run `"start /b /affinity 0x2 nd_send_bw -C <IP_host>"`
- Repeat for port 2 with the appropriate IP.
- 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 and switches. See [RDMA over Converged Ethernet \(RoCE\)](#).
 - CPU/power options are not set to "Maximum Performance".

6.6 Virtualization Related Troubleshooting

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).	<ol style="list-style-type: none"> 1. Boot to BIOS and disable SR-IOV. 2. Burn Firmware with lower number of VFs. 3. Re-enable SR-IOV in BIOS. <p>For more information, please contact Mellanox support.</p>
Running Windows server 2012, 2012 R2 and 2016 as VM over ESX with Mellanox adapter cards connected as Direct pass-through fails to power on.	ConnectX adapter network cards might be trying to use too many MSI-X vectors.	<ol style="list-style-type: none"> 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: <ul style="list-style-type: none"> • In the Name column, add pciPassthru0.maxMSIXvectors. • In the Value column, add 31. 7. Click OK and click OK again. <p>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=10388456420</p>
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.	<p>Please refer to: http://support.microsoft.com/kb/2975719</p> <p>“August 2014 update rollup for Windows server RT 8.1, Windows server 8.1, and Windows server 2012 R2” – specifically, fixes.</p>

Issue	Cause	Solution
The VF adapter is in a 'yellow bang' state, and the following message appears in the host event viewer message log: "Self-Healing second tier policy was activated on Virtual Function (VF) #%%3 with sensors #%%4, and Bus driver restart is needed on the VF. Please restart the Bus driver."	A self-healing reset was activated on the VF, but the driver is in a yellow bang' state and must be manually restarted in order to continue the operation.	Restart the VF driver.
In Hyper-V environment, Enable-Net-AdapterVmq 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.

6.7 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>
- <Device>: <Execution of | Post> FW command failed. op <x>, status <y>, errno <n>, token <m>, in_modifier <k>, op_modifier <l>, in_param <j>.
- <Device>: SR-IOV was successfully enabled. Running in master mode.
- Port type registry value for device <device name> could not be modified to value (PortType = <n>). Previous value will be set.
- <Device>: Virtual Function (VF) #<VF ID> issued an invalid or out-of-sequence device-command (channel=<n>, fragment=<m>, cmd=<k>, status=<l>) - command blocked.

6.8 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 Mellanox-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

6.9 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:

Event Type	Description	Provider	Default	Tag
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_PORT_DN (1 << 2) // i.e. 0x04
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 64 - "Events Causing Automatic State Dumps," on page 214
- 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\CurrentControlSet\Services\mlx4_bus\Parameters and setting it to another value.

7 Appendix: Windows MPI (MS-MPI)

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)

7.1 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`.

7.2 Running MPI

1. Run the following command on each mpi client.

```
start smpd -d -p <port>
```

2. Install ND provider on each MPI client in MPI ND.
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>
```

7.3 Directing MSMPI Traffic

Directing MPI traffic to a specific QoS priority may be delayed due to:

- Except for `NetDirectPortMatchCondition`, the QoS powershell `CmdLet` for `NetworkDirect` traffic does not support port range. Therefore, `NetworkDirect` 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.

7.4 Running MSMPI on the Desired Priority


1. Set the default QoS policy to be the desired priority (Note: this prio should be lossless all the way in the switches*)
2. Set SMB policy to a desired priority only if SMD Traffic running.
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 lossless in 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_SOCKET 1
```

7.5 Configuring MPI

1. Configure all the hosts in the cluster with identical PFC (see the PFC example below).
2. Run the WHCK ND based traffic tests to Check PFC (ndrping, ndping, ndrpingpong, ndpingpong).
3. Validate PFC counters, during the run-time of ND tests, with “Mellanox Adapter QoS Counters” in the perfmon.
4. Install the same version of HPC Pack in the entire cluster. NOTE: Version mismatch in HPC Pack 2012 can cause MPI to hung.
5. Validate the MPI base infrastructure with simple commands, such as “hostname”.

7.5.1 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-netadapter qos -Name
```

7.5.2 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_SOCKET 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_SOCKET 0 -affinity c:\\test1.exe
```

8 Document Conventions and Related Documents

This section describes the conventions, abbreviations and acronyms used in this documentation. It also lists resources for additional related information.

8.1 Document Conventions

Description	Convention	Example
File names	file_name.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	<i>enable</i>
Emphasized words	Italic font	<i>These are emphasized words</i>

8.2 Abbreviations and Acronyms

Abbreviation / Acronym	Whole Word / Description
b	(Lower case) 'b' is used to indicate size in bits or multiples of bits (e.g., 1Kb = 1024 bits)
B	(Capital) 'B' is used to indicate size in bytes or multiples of bytes (e.g., 1KB = 1024 bytes, and 1MB = 1048576 bytes)
EoIB	Ethernet over InfiniBand
FW	Firmware
HCA	Host Channel Adapter
HW	Hardware
IB	InfiniBand
IPoIB	IP over InfiniBand

Abbreviation / Acronym	Whole Word / Description
lsb	Least significant bit
LSB	Least significant byte
MPI	Message Passing Interface
msb	Most significant bit
MSB	Most significant byte
NIC	Network Interface Card
NVGRE	Network Virtualization using Generic Routing Encapsulation
PFC	Priority Flow Control
PR	Path Record
QoS	Quality of Service
RDS	Reliable Datagram Sockets
RoCE	RDMA over Converged Ethernet
SL	Service Level
SW	Software
TC	Traffic Class
ULP	Upper Level Protocol
VL	Virtual Lane
VPI	Virtual Protocol Interconnect

8.3 Related Documentation

Document	Description
<i>MFT User Manual</i>	Describes the set of firmware management tools for a single InfiniBand node. MFT can be used for: <ul style="list-style-type: none"> • 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
<i>WinOF Release Notes</i>	For possible software issues, please refer to <i>WinOF Release Notes</i> .

Document	Description
<i>MLNX OFED User Manual</i>	For more information on SR-IOV over KVM, please refer to the <i>MLNX_OFED User Manual</i> .
<i>InfiniBand™ Architecture Specification</i> , Volume 1, Release 1.2.1	The InfiniBand Specification by IBTA

9 User Manual Revision History

Docu ment Revisi on	Date	Changes
Rev 5.50	Mar. 2019	Converted to online html format; some reorganization of content.
	May 27, 2018	Added Section 3.1.4.8, “How to Configure Storage Space Direct over RDMA”, on page 50
	May 06, 2018	<p>Added the following sections and their subsections:</p> <ul style="list-style-type: none"> • Section 3.1.8, “Receive Segment Coalescing (RSC)”, on page 56 • Section 3.7, “Dump Me Now (DMN)”, on page 134 • Section 3.7.4, “Configuration”, on page 136 (and its sub- sections). • Table 29, “Dump Me Now Configurations,” on page 137 • Section 3.7.3, “Cyclic DMN Mechanism”, on page 136 • Section 3.9.1.1.2, “Nd2AdapterControlSetCqInterrupt- Moderation”, on page 141 • Section 4.6.3.7, “Show selfhealing”, on page 197 • Section 3.10.1.1.1, “Miniport Driver Sensors”, on page 170 • Table 44, “Miniport Driver Registry Keys,” on page 172 • Table 45, “Logging - Windows Event Viewer Applica- tions Messages,” on page 173 • Table 46, “Logging - Windows Event Viewer Messages,” on page 173 • Section 4.6.2.7, “Resources”, on page 192 • Section 4.6.3.9, “show driverParams”, on page 200 • Section 4.6.3.10, “show perfstats”, on page 201 • Table 63, “Virtualization Related Issues,” on page 211 • Section 5.7, “Reported Driver Events”, on page 213 <p>Updated the following sections:</p> <ul style="list-style-type: none"> • Section 1, “Introduction”, on page 18 • Table 5, “Hardware and Software Requirements,” on page 20 • Table 20, “Performance Registry Keys,” on page 120. Added the following note to the “The maximum value of this registry key is 64.” to the *NumRSSQueues registry key. • Section 3.5.3, “Network Virtualization using Generic Routing Encapsulation (NVGRE)”, on page 86 • Table 14, “SR-IOV Mode Configuration Parameters,” on page 101 • Table 27, “RoCE Options,” on page 134 • Table 28, “General Registry Keys,” on page 134 • Table 34, “Mellanox Adapter Diagnostics Counters,” on page 155 • Table 41, “Mellanox Proprietary WinOF Bus Counters,” on page 166 <p>Removed the following sections:</p> <ul style="list-style-type: none"> • Section <i>NVGRE Configuration Scripts Examples</i> and its subsections • Section <i>Configuring NDK for Virtual NICs</i> • Memory Translation Table (MTT) Optimization

Document Revision	Date	Changes
Rev 5.40	May 21, 2017	<p>Added the following sections:</p> <ul style="list-style-type: none"> • Section 3.1.16, “Threaded DPC”, on page 71 (and its sub- sections). • Section 3.9.4.4, “RSS Monitoring”, on page 158 (and its subsections). • Section 3.11.2, “Disabling Mellanox Device while NetworkDirect Applications are Running”, on page 178 (and its subsections). • Section 4.6.2.1, “nonrss-capture”, on page 189 • Section 4.6.2.2, “sw-reset”, on page 189 • Section 4.6.3.1, “show packet-filter”, on page 195 • Section 4.6.3.2, “show qos”, on page 195 • Section 4.6.3.3, “show nd”, on page 196 • Section 4.6.3.8, “show regkeys”, on page 198 (and its subsections) • Section 4.6.3.9, “show driverParams”, on page 200 • Section 4.6.3.10, “show perfstats”, on page 201 • Section 5.1, “RDMA Related Troubleshooting”, on page 204 <p>Updated the following sections:</p> <ul style="list-style-type: none"> • Table 21, “Ethernet Registry Keys,” on page 126 • Table 34, “Mellanox Adapter Diagnostics Counters,” on page 155 • Section 3.9.3, “Tunable Performance Parameters”, on page 150 (and its subsections)
Rev 5.35	January 29, 2017	<p>Added the following sections:</p> <ul style="list-style-type: none"> • Section 3.9.4.5, “Propriety Mellanox Adapter RSS Counters”, on page 162 (and its subsections) • Section 4.6.2.7, “Resources”, on page 192 • Section 4.6.2.8, “Ipoib-ep”, on page 193 • Section 4.6.2.9, “Get-state”, on page 194 • Section 4.6.2.10, “Restart”, on page 194 <p>Updated the following sections:</p> <ul style="list-style-type: none"> • Section 3.5.4, “Single Root I/O Virtualization (SR- IOV)”, on page 88 • Section 3.5.4.1.1, “System Requirements”, on page 88 • Section 3.5.4.1.2, “Feature Limitations”, on page 89 • Section 3.6.3, “Basic Registry Keys”, on page 116 • Section 3.6.4, “Off-load Registry Keys”, on page 118 • Section 3.6.5, “Performance Registry Keys”, on page 120 • Section 3.7, “Dump Me Now (DMN)”, on page 134 • Table 20, “Performance Registry Keys,” on page 120 • Table 34, “Mellanox Adapter Diagnostics Counters,” on page 155 • Table 60, “InfiniBand Related Issues,” on page 206

10 Release Notes Change Log History

Category	Description	Ref. No.
Rev. 5.50.51000 Contains the following versions of components: <ul style="list-style-type: none"> • Bus, eth, IPoIB and mux drivers version is 5.40.14659 • The CIM provider version is 5.40.14659 		
General	Added support for Windows Server 2019.	-
Bug Fixes	See Bug Fixes History .	-
Rev. 5.50 Contains the following versions of components: <ul style="list-style-type: none"> • Bus, eth, IPoIB and mux drivers version is 5.50.14643 • The CIM provider version is 5.40.5.50.14643 		
Dump Me Now (DMN)	DMN is a bus driver (mlx4_bus.sys) feature that generates dumps and traces from various components, including hardware, firmware and software, upon internally detected issues (by the resiliency sensors), user requests (mlxtool) or ND application requests via the extended Mellanox ND API. DMN is unsupported on VFs. For further information, refer to section Dump Me Now (DMN) in the User Manual.	
CPUs	Added supports for systems with up to 252 logical processors when Hyperthreading is enabled and up to 126 logical processors when Hyperthreading is disabled.	1265040
Performance	Added support for RSC solution in TCP/IP traffic to reduce CPU overhead.	598819
	Added support for extended NDSPI to control CQ moderation.	1052685
	Added a new counter for packets with no destination resource.	1078744
	Added a new registry key that allows users to configure the E2E Congestion Control feature.	1125414
	Added to the vlan_config tool the ability to create VLANs for the Physical Function (PF) in addition to the Virtual Function (VF).	1047438
IPoIB	Added support for VMQ over IPoIB in Windows Server 2016.	
Debugging	Added support for collecting firmware MST dumps in cases of system bug check.	1034399
	Added an event log message (ID 273) that is printed when the number of resources to load the VF is insufficient.	1110008
Counters	Added a counter for the number of packets discarded due to an invalid QP number.	1065348
	Added DSCP based counters to support traffic where no VLAN/priority is present.	1117335

Category	Description	Ref. No.
RDMA	RoCE TTL default value has been changed from 128 to 16.	1166268
General	Added support for Windows Server 2019. Note: Currently the drivers in the Windows Server 2019 package are certified only for Windows Server 2016.	
RSC	Modified the RSC default mode when using Windows Server 2019. RSC is disabled by default in Windows Server 2019.	
Bug Fixes	See Bug Fixes History .	
Rev. 5.40 Contains the following versions of components: <ul style="list-style-type: none"> • Bus, eth, IPoIB and mux drivers version is 5.40.13749 • The CIM provider version is 5.40.13749 		
General	Added PDDR (Port Diagnostics Database Register) support: providing details on the root cause when the link is down via mlxtool.	956728
	Improved driver load time by reducing the amount of context initialization work done in the firmware, and performing it in the driver instead.	924738
SR-IOV	Amended the PF\VF counters' wrong data display. For reliable PF/ VF counters, it is recommended to use the WinOf 5.40 version on the host, and WinOf 5.30 or higher as the VF version.	965857
	Modified SR-IOV VF driver to use Mellanox hardware comm channel.	853685
mlxtool	Added the ability to show the sensors state for each instance (bus or miniport)	1044293
	Added the ability to modify receive ring parameters on the fly.	1022471
	Exposed Ethernet driver data path statistics via the mlxtool perfstats command.	1022469
	Added an option to dump all ND connections.	
	Added a command that triggers a "Dump me now" event.	
	Added performing "HCA reset" function.	
	Added the ability to show the difference between actual configured registry keys and values in the registry.	

Category	Description	Ref. No.
RDMA/ND	Extended the ND API to allow reporting to RDMA applications when the device is reset, and when it is back to operational mode. The new ND API header is a part of the SDK.	1020585
		1022739
	Extended the ND API to allow triggering "Dump-me-now" from an ND application for specific QPs.	1022723
	Returning correct return values in case of canceled ND requests.	
	The RoCE Version Interoperability feature is disabled - RDMA communication between nodes, where one node is configured to RoCE v1.5 and the other to RoCE v2, is not supported.	
RSS Monitoring	Added the ability to sample packets which are not sent to the RSS queues via mlxtool.	867203
	Added per-core RSS traffic counters.	867201
ND-IBAL	Removed the option to use or install the ND IBAL provider. In case the ND IBAL provider is installed as part of an upgrade with the full package, it will be removed.	1014850
Troubleshooting	The maximal size of the WPP trace file was increased from 16MB to 100MB.	
Rev. 5.30 Contains the following versions of components: <ul style="list-style-type: none"> • Bus, eth, IPoIB and mux drivers version is 5.35.12978 • The CIM provider version is 5.35.12978 		
Ethernet	Updated driver settings for Virtual Function to receive optimal performance	
	Added RSS (Rx Steering Mode) monitoring counters support	867201
	Added counter for link up/down to count the number of times that the link operative state changes to down. See <i>"Proprietary Mellanox Adapter Diagnostics Counters"</i> in the User Manual.	818135
Tools	vstat tool - Added interface description for each port in the vstat tool.	857255
	Mlxtool - Added support in the following actions: <ul style="list-style-type: none"> • Driver restart • Pulling resource tracker information for VFs 	
		867812

Category	Description	Ref. No.
		756249
Bug fixes	See Bug Fixes History .	
Rev. 5.25 Contains the following versions of components: <ul style="list-style-type: none"> • Bus, eth, IPoIB and mux drivers version is 5.25.12665. • The CIM provider version is 5.25.12665 		
Virtualization	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.	-
	Network Direct Kernel Provider Interface (NDK v2): 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.	-
	SR-IOV Port Mode: Changed the default value of SrioVPortMode to Manual. Now, by default on dual-port devices the maximum number of VFs will be split between the two ports.	
	RDMA over VM in SR IOV Mode (Beta Level): Allows the user to work with ND and NDK over Virtual Machines when in SR-IOV mode.	
Ethernet	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.	-
Bug Fixes	See Bug Fixes History .	-
Rev. 5.22 Contains the following versions of components: <ul style="list-style-type: none"> • Bus, eth, IPoIB and mux drivers version is 5.22.12433. • The CIM provider version is 5.22.12433 		
VXLAN	Setting the dynamic VXLAN UDP port is now supported for dual-port devices when only a single port is active.	-
RDMA	Improves cache hit rate in RDMA by reducing the size of the Adapter's Memory Translation Table (MTT).	-

Category	Description	Ref. No.
	Changed the ND port allocation scheme from hashing to 64k bitmask.	-
	Changed the default value of RoCE mode to RoCE v2	753974
Tools	Modified the Vsat tool to function also when RoCE is disabled.	-
	nd_write_bw and nd_send_bw now support getting send completions using events instead of polling by using -e switch. Parameter only affects client side and only on “duration” mode (-D <time>).	665164
	Improved mlxtool error handling for the pkeys option. When a broken IPoIB interface registry entry exists (for example, an old virtual interface that was not fully removed), the tool would fail and exit. Now the tool skips such entries and prints the next ones.	642352
General	Enabled dual-port card to work as a single-port card.	-
Diagnostic	Improved Event Log Messages explaining behavior of the driver in case of illegal port configuration Port1: Ethernet w/o RoCE, Port2 IB.	681229
	Improved Event Log Messages issued on driver-generated dumps.	648731
SR-IOV	SR-IOV is disabled when the port is set as IB type.	-
	Set the *PriorityVLANTag registry key of Virtual Function (VF) to not support neither priority nor VLAN, and removed VlanId registry key completely.	659090
	[Beta] Added support for SR-IOV Ethernet Windows over KVM.	-
Installation	Removed the co-installer that installs performance counters. The counters is installed during the setup.	-
Teaming	Added support for tagged and untagged traffic over Team driver.	-
	Enabled teaming configuration via the Command Line Interface.	-
Documentation	Release Notes and User Manual documents were removed from the package. A new README file which includes basic installation instructions, summary of main features and requirements has replaced them.	661394

Category	Description	Ref. No.
Rev. 5.19.11822 (Beta Level) Contains the following versions of components: <ul style="list-style-type: none"> • Bus, eth, and IPoIB drivers version is 5.19.11822 • The CIM provider version is 5.19.11822 		
SR-IOV	Added VLAN support for NDK on VF.	
Ethernet	Added a thread race protection for RX/TX CQ/ring iterator	
Rev. 5.19.11803 (Beta Level) Contains the following versions of components: <ul style="list-style-type: none"> • Bus, eth, and IPoIB drivers version is 5.19.11803 • The CIM provider version is 5.19.11803 		
NDKPI	Added support for NDKPI v2.0 interface.	
Virtualization	Added support for RoCE in SR-IOV VM. Virtualization: Added support for RoCE in virtualization mode in the hypervisor.	
PacketDirect	Added support for PacketDirect Provider Interface (PDPI).	
SR-IOV Security	Added support for the Ethertype spoof protection feature, which enables the hypervisor to control the allowed Ethernets that the VF can transmit.	
VMMQ	Added support for RSS load-balancing offload in HW for non-SRIOV VMs.	
Rev. 5.10 Contains the following versions of components: <ul style="list-style-type: none"> • Bus, eth, IPoIB and mux drivers version is 5.10.11345. • The CIM provider version is 5.10.11345 		
Operating System	Added support for a new Windows Client version - Windows 10 Client	-
General	Fixed an issue where a device state dump used for debug could cause the device to get stuck, requiring driver restart or server reboot to recover.	568240
	Fixed an issue where a system with two or more adapters could crash when one of the adapters gets disabled.	532481
	Fixed the case where during very high CPU load on the core that is in charge of transmit notification processing, driver may mistakenly decide the device is hung and initiate a reset to the interface.	550016

Category	Description	Ref. No.
	Fixed the issue of when a driver fails to start (Mellanox device appears with Yellow bang in the device manager) and user stops the driver, this could cause the system to hang.	492885
General	Added third party branding for ATTO. ATTO devices will be shown in the device manager with customized ATTO device names and model numbers.	520073
RDMA	Fixed synchronization issue between client and server side in nd_write_lat that could cause the test to hang on start.	559668
	Fixed a problem of when a user is trying to change the RoCE mode using the Set-MlnxDriverCoreSetting PowerShell command on a setup with two devices, the mode does not change until the next driver restart.	520406
	Enabled the driver to use a minimal number of memory registration resources when physical memory contiguity allows it.	557943
	Implemented a resource pool to save physically contiguous memory in the driver, which was used by RDMA applications. This way the re-use of this memory will be allowed.	557954
RDMA	Improved ND connection establishment time when using firmware v2.34.5000	495620
	Changed NDK and its clients (like SMBDirect) to be disabled by default when PFC is not enabled.	449771
	Added an RDMA test to the package to demonstrate Linux interoperability.	541340
	Enlarged private data limit in ND and NDK.	562879
	Changed default RoCEv2 UDP destination port to match IANA standard.	574918
	Removed all deprecated performance tools.	569889
QoS	Fixed an issue when after OS initiated reset of the interface, default QoS policies were not properly restored.	558513
Virtualization	Fixed the issue of when enabling VMQ after SRIOV has been disabled, VMQ would not work, and the VM would fall back to paravirtualization, impacting performance.	549092
	Improved isolation of SRIOV host from VM driver issues.	549073

Category	Description	Ref. No.
	Fixed a race which caused some of the configurations to be badly reinitialized during VM or host transition to VMQ mode. This bug used to cause loss of communication to the affected VM or host.	560789
	Added support in mlxtool to allow the query of PKeys configured in SR-IOV VMs.	565011
RoCE	Fixed an issue where adding VLANs would fail because the driver's internal table was not cleaned up correctly upon VLAN removal.	547762
	Fixed a memory leak caused by a race between successful finish of a Management Datagram (MAD) and canceling it.	541447
Debugging	Enabled mlxtool to allow a manual generation of register dumps.	542828
	For performance tuning purposes, debug counters were added to indicate once the driver transmit queue got full.	-
	Added a discard counter for performance analysis.	565011
	Added discard counters per Transport Class. The counters names are in beta and are subject to change.	591908
Ethernet	Fixed the issue when after removing the Virtual Ethernet Adapter, some registry keys would remain.	529621
	Enabled a device watchdog mechanism that prevents the device from sending excessive pauses to the network for any reason.	
	Added support for Windows 2008R2/Windows 7 Client teaming driver to allow selecting the MAC address of the primary interface.	514256
	Added support for driver Teaming in Windows Client 8.1.	507319
InfiniBand	Fixed the issue of when Query Path Record (QPR) option is set, a race condition occurs. The race would be between the handling of received packets and the response to the path query to the new destination. This could cause losing packets received from a new source because the path query for it was not yet finished.	536405
	Fixed the Query Path Record list handling to prevent double entering of the same destination. This bug may have caused list corruption which led to unexpected results.	535446
	Fixed the issue of when BSOD may occur when running with two HCAs and using sminfo when no Subnet Manager is available.	492579

Category	Description	Ref. No.
	Fixed the issue of when creating virtual IPoIB PKey interfaces with HP cards using part_man.exe utility was not possible.	491585
	Fixed the issue where a new VM creation or its migration in IPoIB could cause the system to crash.	441213
Performance	Fixed an issue where in VMQ mode, not all receive buffers allocated for the VMQ are used, impacting performance.	567513
Installation	Added support for installing counters with co-installer. This allows the installation of counters while installing the driver via the INF mechanism.	549805
Rev. 4.95.50000 Contains the following versions of components: <ul style="list-style-type: none"> • Bus, eth, IPoIB and mux drivers version is 4.95.10777 • The CIM provider version is 4.95.10777 		
InfiniBand	Fixed BSOD on next driver restart when running the sminfo tool and SM is not running	492579
	Fixed instability in IPoIB driver when polling mode is enabled	521205
	Fixed the issue when live-migrated VM with virtual NIC over IPoIB physical interface loses its IP address and comes up with APIPA address (169.254.x.x)	439359
Ethernet	Fixed BSoD after the receive buffer's size changes in VMQ mode	500228/ 522073
	Fixed Powershell setting of RoCE mode when the machine has more than one Mellanox device	520406
	Fixed reporting of NVGRE capabilities to the OS	535203
	Added a new mode that ignores FCS warnings and enables the Ethernet packets to be received by the NIC	-
	Added the option of setting the MAC address of 2008R2 teaming driver to be taken from the primary interface	-
	Added the option of taking the MAC Address used for teaming from one of the interfaces without manipulation	-
Changes in UI	Added the option of configuring the team MAC address to be the same as the primary adapter MAC address	514989

Category	Description	Ref. No.
Troubleshooting	Added an autologger session that dumps WPP traces to file to enable easier analysis of issues	-
	Added logging of performance counters and CPU power saving state to system snapshot tool	-
RDMA	Fixed handling of send request with inline data larger than supported	443355
	Added the option to allow RDMA programmers to create pre-allocated pools of ND resources to reduce resource creation time	-
Low Level Driver	Fixed the case in which the system rarely gets unstable after driver restart	492885
Infrastructure	Updated customization for OEM cards	-
Rev. 4.90.50000 Contains the following versions of components: <ul style="list-style-type: none"> • Bus, eth, IPoIB and mux drivers version is 4.90.10714 • The CIM provider version is 4.90.10714 		
Generic	Fixed driver instability when handling many RDMA connection requests in parallel	461854
	Added to MLNX_System_Snapshot Mellanox specific counters and data from Get-Mlnx* Cmdlets	467529
Resiliency	Reset Flow improvements: <ul style="list-style-type: none"> • Resolved race condition when reset is initiated by more than one source • Reset initiated on one port does not cause reset of the other port 	400887
Ethernet	ETS is now configurable through DSCP values. For further details, please refer to WinOF User Manual, "Differentiated Services Code Point (DSCP)" section	434105
	Fixed the issue of when creating a Virtual Ethernet Adapter interface and removing it immediately a Blue Screen may appear	456279
	Fixed duplicated values of Receive Completion Method in Advanced Properties driver dialog on Windows Server 2012 R2	443273
	Performance Improvement: Reduced memory access time for Receive descriptors	-
	VM Scalability: More efficient handling of VMQ control path in HyperV	-

Category	Description	Ref. No.
	Reduced the amount of kernel memory used for each Ethernet interface by the driver	-
	Virtual Machine traffic on the default queue now uses a single CPU core as required by Microsoft. This applies both to SRIOV and VMQ	441581
InfiniBand	Updated IBAL interface version. In order for the applications that use the IBAL interface to work with WinOF Rev 4.90.50000, they must be recompiled with the new SDK	-
	Added support for SM change event	435564
	Fixed propagation of error code when ib_join_mcast() fails	448028
	Fixed connectivity problems when using PKeys from the same partition with different membership types	417753
	Fixed VM reset after printing the message "mlx4_core 0000:05:00.0: unparavirt command: OTHER (0x3a) accepted from slave:3" in SR-IOV InfiniBand VM over non-windows hypervisor	422598
RoCE	In RoCE v2, added the option of determining the source port field of the UDP header by the application	-
NDK	Improved CPU utilization by changing ndkgetremotetokenfrommr() to return value in network byte order	-
Performance	Fixed the UI crash when working with a single port	427484
	Increased the accuracy of the run time duration parameter of ND Performance tests even when sending large message	-
	Fixed Maximum value for ThreadPoll parameter to be 200,000, instead of the previous state when it could not be set above 20,000 due to a bug	481291
IPoIB	Fixed stability issues	-
	Fixed displaying of IPoIB default turning option	428601
	Fixed temporary network connectivity issues while migrating VMs or modifying VMQ configuration for VMs that uses IPoIB with VMQ	417687
	Fixed the part_man tool to use the actual default p_key instead of 0xffff	417858

Category	Description	Ref. No.
	Fixed NIC reset when attaching to a multicast group fails	423435
	Fixed duplicated values of Receive Completion Method in UI on Windows Server 2012 R2	-
	Added support for multiple PKey interfaces in IPoIB	-
	Added support for teaming of IPoIB interfaces to allow failover	443273
	Added IPoIB adapters teaming support (beta level)	-
	Added sending of gratuitous ARP in IPoIB interface when the MAC address is changed	408388
	Reduced memory footprints of IPoIB interfaces	-
	Reduced the multiple number of path record queries to one when old query information exists	466336
	Improved completion memory access speed	440018
	Changed default VMQ/VPort affinity to use first RSS CPU	442549
	Multiple PKey support is now at GA level. The part_man tool allows the creation of up to 64 vIPoIB interfaces (32 per port)	-
IPoIB	Added a warning to the event log if the port MTU is higher than the reported MTU by the SM.	-
Installation	Fixed CIM failure after installation in maintenance mode	423206
	Fixed loading of old driver after driver upgrade that requires system reboot to complete the process	422812
	Fixed RoCE disable by default after installation of WinOF in Windows 8.1 Client	454020
ND	Fixed seg fault when executing ND application with no device installed or when a wrong device identifier is used	431113
	Fixed wrong reported value of supported number of SGE in 32 bit DLLs	425841
	Increased the number of supported SGEs in 32 bit DLLs to 2	425841

Category	Description	Ref. No.
NVGRE	Fixed restoration of NVGRE configuration after NIC reset	442478
Changes in UI	Replaced the terms “LBFO” and “Bundle” with “Teaming” and “team” respectively.	-
CIM/WMI	Added support to query/set/enable/disable ECN	
	Added support to query DroplessMode state	
	Fixed the issue when using the PowerShell command GetMlnxFirmwareIdentity on a system with multiple NICs/HCAs while one of the devices is disabled and the command fails	
Rev. 4.80.50000 Contains the following versions of components: <ul style="list-style-type: none"> • Bus, eth, IPoIB and mux drivers version is 4.80.10388 • The CIM provider version is 4.80.10388 		
Installation/ Upgrade	Added check for administrator privileges during installation	391704
	Added support for installation in silent mode without execution of perf_tune	397946
	Fixed installation stuck when Remote Desktop Session Host Windows Installer RDS compatibility is enabled	371541
Generic	Changed Reset Flow (+SR-IOV)-enabled only if no user space application is running and depends on the registry key: AllowResetOnError setting)	370536
	Changed the number of supported QPs in a multicast group from hard coded value to firmware capabilities dependent	401850
	Fixed driver load failure in machines with 1 TB memory and above	407556
	Fixed memory leak on the Virtual Machine in SR-IOV when resetting the Virtual Machine of associated VFs	373144
IPoIB	Added multiple P_Key support (beta level)	391240
	Added IPoIB SR-IOV over KVM and ESX Hypervisors (for both full and partialmembership)	-
	Added support for LID change event	-

Category	Description	Ref. No.
	Added enhancements in part_man for the multiple Pkey support	-
	Changed IPv6 “all dhcp servers” mcast to be persistent	-
	Fixed rare cases of driver hang following a Subnet Manager failover event	-
	Fixed stability issues	-
Ethernet	Added RSS in UDP (enabled by default)	-
	Added 56 GbE (Please refer to the Infiniband Switch User Guide for further details)	-
	Changed DSCP configuration to be per port instead of global	394703
	Network Direct: Fixed race in NDK between handling of incoming connection and destruction of a listener	-
	Network Direct: Fixed race between NDK object creation and usage	-
	Improved TCB (Transmission Control Block) management on send	389974
	Improved transmit and receive in multi stream scenarios	-
	Enabled hardware checksum offload for non TCP/UDP traffic with ConnectX®-3 Pro	394977
	Improved stability when handling OIDs during driver reset	-
	Fixed performance tuning for 1GbE link	-
	Fixed possible reset of driver during migration of large number of VMs at the same time	401655
	Fixed stability issues	-
RoCE	Added RoCE IP based	391238
ND	Fixed wrong return value in IND2Adapter::QueryAddressList	-
InfiniBand	Added non-default PKey in VM	-

Category	Description	Ref. No.
Performance	Optimized interrupt moderation values in SR-IOV VF mode for IPoIB	-
	Improved perf_tuning detection for the first port	-
	Improved performance in packet forwarding scenarios	-
	Decreased dropped packets rate for Ethernet significantly	414872
	Changed default perf_tuning scenario to be “Balanced configuration”	396981
	Various performance improvements	-
WMI/CIM	Added ability to read active RoCE configuration from hardware	400598
	Added support for RoCE IP Based	390573
Rev. 4.70.50050 Contains the following versions of components: <ul style="list-style-type: none"> • Bus, eth, IPoIB and mux drivers version is 4.70.10143 • The CIM provider version is 4.70.10143 		
IPoIB	Fixed SM fail-over causing the driver to hang	-
Rev. 4.70.50040 Contains the following versions of components: <ul style="list-style-type: none"> • Bus, eth, IPoIB and mux drivers version is 4.70.10141 • The CIM provider version is 4.70.10141 		
Generic	Optimized handling of “affinity change” on OID_RECEIVE_FILTER_QUEUE_PARAMETERS	-
	Added the ability to control the number of retries and timeout to check the device health before performing reset	-
Ethernet	Fixed missing pause response by sender when using DSCP/untag priority tag mode with ETS enabled	-
Rev. 4.70.50000 Contains the following versions of components: <ul style="list-style-type: none"> • Bus, eth, IPoIB and mux drivers version is 4.70.10126. • The CIM provider version is 4.70.10130 		

Category	Description	Ref. No.
Installation/Upgrade	Fixed removal of virtual IPoIB ports in uninstallation	-
	All user-space binaries are now signed	-
	Fixed restoration process of DNS servers during upgrade	-
	Fixed popping windows during installation/upgrade	-
	Fixed missing 32 bit files in the catalog files	-
Generic	Changed Ethernet and IPoIB event log messages to be more clear	-
	Ported SDK project to Visual Studio 2013.	-
	Fixed an issue which caused Mellanox miniport devices to be listed in “Devices and Printers”	-
	Fixed Ethernet and IPoIB deadlock in power state change during shutdown/reboot	-
	Fixed stability issues	-
IPoIB	Added support for IPoIB SR-IOV Virtual Function (VF) over KVM Hypervisor (Beta level)	-
	Added support for non-default pkey, as queried from OpenSM, on IPoIB SR-IOV VF over KVM.	-
	Added IPoIB QoS proprietary counters, diagnostics and traffic for monitoring, using Windows' Perfmon utility	-
	Fixed part_man exit with return value 0 in case of error	-
Ethernet	Added support for Ethernet SR-IOV over Windows Hyper-V Hypervisor (over Windows Server 2012 R2)* * Requires firmware v2.30.8000 and above	-
	Added Virtual Ethernet Adapter support which enables using SMB Direct and HyperV (VMQ and NVGRE (over ConnectX®-3 Pro)) on the same port** ** Requires firmware v2.31.5050 and above.	-
	Added lossless TCP buffer management when no receive WQE are available	-

Category	Description	Ref. No.
RoCE	Added ConnectX®-3 Pro support for RoCEv2	-
	Changed the transport name in vstat and ibstat to be RoCE v2.0	-
	Fixed ibstat behavior on devices with RoCE enabled	-
	Fixed releasing of RDMA resources and reacquire them on power down and up.	-
	Fixed RDMA Activity counters which didn't increase for ND traffic	-
ND	Fixed hard-coded limitation of 4 SGEs	-
InfiniBand	Fixed vstat printing of counters for Ethernet ports	-
	Fixed crash when calling ib_join_mcast() with timeout_ms = 0	330740
Performance	Improved perf_tuning setting in single CPU machines to avoid TX collision	-
Rev. 4.61 (Released as an intermediate release)		
Installation/Upgrade	Fixed an issue preventing JumboPackets registry key to be restored correctly	-
	Ensured that uninstallation of Mellanox package in Virtual Machine leaves the system clean	-
Generic	Improved information in event log when a bad cable is detected	-
	Improved resiliency on error flow in Ethernet, IPoIB and bus drivers	-
	Fixed an issue which caused Mellanox devices to be listed in “Devices and Printers” and had “Safe Removal” UI	-
Performance	Added support OF IPv6 to all nd_*_* tests	-
	Enabled optimal interrupt moderation values in SR-IOV VF mode	-
	Stopped using NdisQueryNetBufferPhysicalCount to improve CPU utilization	-
IPoIB	Enabled searching for IBAT routes based on dest only instead of src,dest and added a mechanism preventing memory growth in IBAT entries	-

Category	Description	Ref. No.
	Allowed any number of RSS processors, not only a power of 2	-
	Ensured SR-IOV mode is not enabled for IPoIB ports, which resulted in confusing message in event log	-
	Fixed error statistics collection which could cause false error report	-
	Fixed a connectivity problem between Hyper-V VMs on the same host	-
	Fixed loopback issues in the virtualization environment	-
	Fixed stability issues	-
Ethernet	Added support for “unknown” link state indication	-
	Added support for DMA checks by driver verifier on SR-IOV Virtual Function	-
	Added support for NVGRE over LBFO Team	-
	Improved performance of handling change receive ring affinity request	-
	In SR-IOV mode, improved resiliency to driver failures in the Virtual Machine which could result in driver load failure in VM	-
	In SR-IOV mode, improved resilience in VF to PF communication	-
	Improved structure of INF file for SR-IOV Physical and Virtual Functions	-
	Fixed an issue that prevented receiving ARP traffic in NVGRE mode	-
Rev. 4.60.17718 Contains the following versions of components: <ul style="list-style-type: none"> • Bus, eth, IPoIB and mux drivers version is 4.60.17718. • The CIM provider version is 4.60.17718. 		
Hyper-V	Fixed NIC reset when moving IPoIB interface in a VM from non-VMQ to VMQ or from VMQ to non-VMQ	325607
Installation/Upgrade	Enabled configuration changes saving upon Inbox and previous releases upgrade	-
	Enabled CIM installation as a standalone package	-

Category	Description	Ref. No.
	Fixed an issue occurred when uninstalling and reinstalling the driver. The ConnectX-3 Pro Ethernet device was displayed in the Device Manager with a yellow bang (!).	-
	Fixed an issues enabling the package's execution in modify mode resulting in driver being disabled	-
Generic	Added support for a new report for bad cables *** *** Requires firmware v2.30.8000 and above.	-
	Fixed random parsing failures of string registry entries	-
	Fixed compilation failure of "Hello_world" in the SDK	-
	Fixed the return value of <code>ib_query_ca()</code> if failed to allocate resources for operation	-
Performance	Added support to IPv6-to-all <code>nd_*_*</code> tests	-
	Fixed CPU utilization report in <code>nd_*_*</code> tests	-
	Fixed correct bandwidth peak results in <code>ibv_send_bw</code> with UD QP	-
	Fixed sync problems of bidirectional mode in <code>ibv_read_bw/ibv_write_bw</code>	-
	Fixed an issue reporting incorrect adapter type in performance tuning log file	-
RoCE	Fixed RoCE mode parsing	-
ND	Added the ability to rearm a CQ in the kernel	-
	Added the ability to handle LID changes	-
	Changed connection timeout behavior. Added the <code>STATUS_CONNECTION_REFUSED</code> return value upon connection timeout.	-
	Fixed missing completions when working with Completion Queue with single entry	-
IPoIB	Added the ability to handle LID changes	-
	Added support for iSCSI boot over IPoIB	-

Category	Description	Ref. No.
	Fixed unexpected behavior upon QP asynchronous event	-
	Fixed bad completions of VMQ and NonVMQ modes in IPoIB	-
	Fixed a failure occurred when setting the IPoIB adapter value to “SA Query Timeout”	-
	Fixed propagation of the physical link disconnection to virtual (part_man) interface	-
	Fixed BSOD caused by calling ib_join_mcast() with timeout_ms = 0	-
	Performance improvements in latency	-
Ethernet	Added DSCP support over IPv4a	-
	Added traffic profile	-
	Added IRQ dynamic moderation	-
	Modified the CQ size to prevent CQ overrun	-
	Changed the report link speed zero in case of disconnected network adapter	-
	LBFO: Fixed port channel teaming with CISCO switch and Fabric Extenders traffic loose in Windows Server 2008 R2	-
	Fixed an issue related to packets sent with corrupted VLAN header when they were meant to be untagged	-
	Fixed unexpected behavior upon QP asynchronous event	-
	Fixed the ability to disable Wake-on-Lan (WoL) on NICs which supports it.	-
	Stability fixes	-
	Performance improvements	-
WMI/CIM	Added ControlledBy association to IBPort	-
	Fixed ConformsToProfiles association for SoftwareIdentity and DriverIdentity	-

Category	Description	Ref. No.
	Fixed execution of all tests which were running when executing Diagnostic tests on one instance	-
	Fixed a failure occurred when running MLNX_Card	-
	Fixed the printing of diagnostics log	-
	Fixed an issue preventing from get-event to show information after disabling the PCI device	-
	Removed support for the following configuration: <ul style="list-style-type: none"> • ModeFlags • SingleMsixNum • MultiMsixNum • SingleEqNum • MultiEqNum • MaxContQuant • SlaveNum • DebugLevel • DebugFlags • UsePrio • NumFcExch • EnableQoS • BlockMcastLoopBack • InterruptFromFirstPacket • ProbeVf 	-
Rev. 4.60.17738 Contains the following versions of components: <ul style="list-style-type: none"> • Bus and eth driver version 4.60.17718. • The CIM provider version is 4.60.17718. • The mux driver version is 4.60.17729. • The IPoIB driver version is 4.60.17736. 		
IPoIB	Fixed using CQ after VMQ is closed	-
	Fixed bad completion of VMQ QP that was caused by malformed WR	-
Ethernet	LBFO: Fixed the team's MAC address uniqueness in the subnet of the team in Windows Server 2008 R2	-
Rev. 4.55		

Category	Description	Ref. No.
Generic	<ul style="list-style-type: none"> Added support for Windows Server 2012 R2 Operating System Added the ParentBusPath option to each port registry key Added a new hardware ID for ConnectX®-3 Pro NICs The QP numbers allocation is now round-robin manner RecvCompletionMethod as Interrupt is no longer supported Removed the LsoV1IPv4 from the registry/UI Removed from the bus driver configuration the 'Non-DMA' option Removed the TXRingNum option from the UI 	-
NVGRE	<ul style="list-style-type: none"> Added NVGRE hardware off-load support (for ConnectX®-3 Pro cards only) Added to the UI the *EncapsulatedPacketTaskOffload option when using ConnectX®-3 Pro NICs 	-
Performance	<ul style="list-style-type: none"> Added the nd_send_bw and nd_send_lat ND benchmarking tools Fixed nd_*_bw to achieve better performance (memory buffer alignment) and consistent results 	-
Ethernet	<ul style="list-style-type: none"> Fixed the issue preventing messages to be sent in VLAN 0 when using many VMQ rings Added IP-IP checksum off-load support Added Ports TX arbitration/Bandwidth allocation per port The following ND providers, MLX4ND and MLX4ND2 are installed by default Fixed setting the correct SL in UD traffic over RoCE 	-
InfiniBand	<ul style="list-style-type: none"> IPoIB performance improvements Fixed a part_man issue related to wrong statistics over virtual partman interfaces 	-
RoCE	<ul style="list-style-type: none"> Enabled roce_mode value overwrite in case it exists during installation Fixed in ibv_devinfo the display of correct transport RoCE mode Added Sniffer for RoCE packets The used RoCE mode set upon driver load is printed into event log message 	-
Rev. 4.40		
Generic	<ul style="list-style-type: none"> Added a notification in the event log in case SMB is not supported in ConnectX®-2 firmware Added the trace tool for WPP tracing Added copyright to the SDK files Added WMI/Powershell support Fixed an issue causing the setup to fail upon perf_tuning failure during the installation. An error message will be printed in the installation log upon perf_tuning failure. Removed port setting registry key during uninstall Fixed an issue with the Mellanox adapter being shown on the USB removal menu, which caused the removal of the Mellanox adapter once removing the USB. 	-

Category	Description	Ref. No.
Performance	<ul style="list-style-type: none"> • Set 512 RX buffers by default • Removed TXRingNum • Changed the perf_tuning setting to achieve a better performance tuning • Added the nd_write_bw/nd_write_lat and nd_read_bw/nd_read_lat tools • Fixed the perf_tuning indication of the last chosen tuning scenarios • Fixed a crash in the ib_send_lat/bw utilities caused when the port link was down • Fixed the “Restore to defaults” option in the perf_tuning tool. Now the default values are being restored 	-
Ethernet	<ul style="list-style-type: none"> • Added Transmit Side Scaling (TSS) • Added Ethernet QoS proprietary counters, diagnostics and traffic for monitoring, using Windows’ Perfmon utility • Added to the MTU size the IP header size (1500->1514, 9600->9614). Thus the minimum Jumbo frame size is 614. • Interrupt moderation supports the following profiles: <ul style="list-style-type: none"> • Low Latency • Moderate • Aggressive In addition to old values that are not supported anymore. 	-
	<ul style="list-style-type: none"> • Made mlx4_bus and Ethernet devices removable • Network Direct: Added support for NDv2 • Network Direct: Set the default ND provide value to mlx4nd2 • Fixed WoL support on NIC with a single port • Fixed the default RoCE configuration on NICs with a single ports • Fixed the values for the MTU and rate of the CM-REQ • Fixed miniport reset on sending scenarios • Removed the QoS attributes when disabling QoS 	

Category	Description	Ref. No.
Ethernet	<ul style="list-style-type: none"> Enabled MaxRssProcessors support of the following values: 1, 2, 4, 8, 16, 32, 64 Network Direct: Fixed a crash occurred when more than 4 SGEs elements were used in an ND write operation Network Direct: Fixed the swap of InboundReadLimit and OutboundReadLimit when creating an EndPoint and in Connector::GetConnectionData Network Direct: Fixed disallowing creation of EndPoint with zero attributes in the Receive Queue Network Direct: Removed the option of NDK registration failure requiring a reboot of the machine to register it again Network Direct: Fixed a failure when creating an EndPoint with zero attributes in the Receive Queue Network Direct: Added the option of sensing the incoming Read messages according to the device capabilities when creating an EndPoint limit Network Direct: Fixed a failure of ND connectivity between VMs on the same host Added Transmit Side Scaling (TSS) Added Ethernet QoS proprietary counters, diagnostics and traffic for monitoring, using Windows' Perfmon utility Added to the MTU size the IP header size (1500->1514, 9600->9614). Thus the minimum Jumbo frame size is 614. Interrupt moderation supports the following profiles: <ul style="list-style-type: none"> Low Latency Moderate Aggressive In addition to old values that are not supported anymore. 	-
Ethernet	<ul style="list-style-type: none"> Made mlx4_bus and Ethernet devices removable Network Direct: Added support for NDv2 Network Direct: Set the default ND provide value to mlx4nd2 Fixed WoL support on NIC with a single port Fixed the default RoCE configuration on NICs with a single ports Fixed the values for the MTU and rate of the CM-REQ Fixed miniport reset on sending scenarios Removed the QoS attributes when disabling QoS Enabled MaxRssProcessors support of the following values: 1, 2, 4, 8, 16, 32, 64 Network Direct: Fixed a crash occurred when more than 4 SGEs elements were used in an ND write operation Network Direct: Fixed the swap of InboundReadLimit and OutboundReadLimit when creating an EndPoint and in Connector::GetConnectionData Network Direct: Fixed disallowing creation of EndPoint with zero attributes in the Receive Queue Network Direct: Removed the option of NDK registration failure requiring a reboot of the machine to register it again Network Direct: Added the option of sensing the incoming Read messages according to the device capabilities when creating an EndPoint limit Network Direct: Fixed a failure of ND connectivity between VMs on the same host 	

Category	Description	Ref. No.
InfiniBand	<ul style="list-style-type: none"> On rare occasions, depends on the GUID assignment, the IPoIB MAC address can be assigned with a multicast MAC (the least significant bit of the most significant address octet is set to 1). In that case, all of the traffic over the IPoIB I/F is dropped. <p>If you experience this issue, please contact Mellanox support.</p> <ul style="list-style-type: none"> Added active_mtu field to struct ib_port_attr_t Added the option of vstat displaying the active_mtu of the ports Allowed registration of a large Memory Region which is splitted to many segments Fixed a bluescreen issue that occurred when disabling the interface after a TX stress over the VMQ Fixed a failure of MPI/ND over InfiniBand Added the option of ibv_devinfo displaying the correct MTU value after it was changed Added the option of part_man printing the adapter name when the Port GUID is set to zero. Added the option of part_man printing the leading zeroes of port GUID 	
Installation/Upgrade	<ul style="list-style-type: none"> Prevented displaying a message to upgrade the firmware for OEM NICs if it has the latest firmware version Removed portsetting registry key during uninstallation 	
Rev. 4.3 (This version was released as an intermediate release)		
Generic	<ul style="list-style-type: none"> Added support for a new provider called MLX4ND, which supports both NDv1 and NDv2 interfaces 	
Performance	<ul style="list-style-type: none"> Enabled performance tuning running according to the operating systems that are running over it. <p>The keywords added to the registry in NDIS support Windows Server 2012 are:</p> <ul style="list-style-type: none"> RssMaxProcNumber 	
	<ul style="list-style-type: none"> NumRSSQueues RSSProfile <p>The rest of the keywords are added in all versions of NDIS. This change is based on: http://msdn.microsoft.com/en-us/library/windows/hardware/ff570864(v=vs.85).aspx</p>	

Category	Description	Ref. No.
Ethernet	<ul style="list-style-type: none"> RoCE MTU value is no longer set to 1024 by default. <p>All options stay as they are and can only be chosen if they were selected explicitly in the UI/registry.</p> <p>The current default state is as follows: The value is now derived from the MTU (or MaxFramSize, or Jumbo Packets value) and they are all aliases for the same value). The value is aligned to 256,512,1024,2048 in a way that it will be rounded down to the nearest power of two of the ETH MTU.</p>	
InfiniBand	<ul style="list-style-type: none"> Added ibdiagnet utility support 	
Rev. 4.2		
Generic	<ul style="list-style-type: none"> Modified RSS cores and changed VMQ affinity on the fly Fixed restart issue when there are not enough MSI-X vectors for each machine core Added support for K-GROUPS processors (more than 64 processors support) to allow assignment of MSI-X affinity for multiple processor groups. Set an adequate number of MTTs to map all physical memory Allocated firmware and ICM memory in chunks of non-paged memory instead of using contiguous physical memory. Fixed RSS indirection table mapping building when there are less RX rings than RSS cores. Fixed a bug, preventing standard work with BAR value more than 4GB. Fixed memory leaks Fixed error flows causing a Bluescreen in driver startup/unload Fixed a Bluescreen occurrence upon shutdown due to leak in active resources 	
Generic	<ul style="list-style-type: none"> Changed device names in device manager and their hardware IDs. The changes were made to distinguish between ConnectX®-2 and ConnectX®-3: <ul style="list-style-type: none"> for ConnectX-2: MLX4\ConnectX-2_Eth and IBA\ConnectX-2_IPoIB 	
	<ul style="list-style-type: none"> for ConnectX-3: MLX4\ConnectX-3_Eth and IBA\ConnectX-3_IPoIB Set QoS settings only for ConnectX-3. Changing the hardware ID, forces the OS to install new device and re-build the registry keys. Added an event log to indicate driver failure upon start if there are two HCA burned with the same GUID. Added firmware upgrade support as part of the setup process. The setup burns the new firmware only on Mellanox cards. Firmware burning failure does not prevent the driver's installation, therefore, it will show a warning. In this case, it is recommended to update the firmware manually. Enabled configuration of TxRingNum registry key from the UI Improved the "Port Protocol" dialog Added Registry key documentation to the setup package 	

Category	Description	Ref. No.
Performance	<ul style="list-style-type: none"> • Optimized code performance • Increased send parallelism • Memory used in receive flow is now allocated with the same affinity of the handling processor for faster access • Statistics parameters are now directly read from hardware instead of being calculated by software. • Added support for BlueFlame. BlueFlame is now the default working mode for all packets that have a descriptor which fits into a BF register (currently 256 bytes). Use "BlueFlame" registry key to enable/disable this feature. • Added support for RSS functionality on available processors numbers. Used to be restricted to start at the first processor. • Changed RSS registry defaults to give better out of the box performance • Added a performance UI to tune performance under various scenarios • Added a tool to tune performance under various scenarios 	
Ethernet	<ul style="list-style-type: none"> • Added support for multiple TX rings • Added an option to verify that the number of multicast groups used is no higher than the firmware limits • Improved performance in virtualization when using VMQ 	

11 API Change Log History

Release	Name	Description
5.40.50000	ND IBAL provider	Disabled ND IBAL provider.
	Extended ND API	Allows reporting to RDMA applications when the device is reset, and when it is back to operational mode. For more information see the User Manual.
4.95.50000	ND extension for Resource pools	Please refer to MLNX_VPI_WinOF_User_Manual_v4. 95
4.80.50000	RDMA_TRANSPORT_RDMAOE_1	It is an alias to: RDMA_TRANSPORT_RDMAOE
	RDMA_TRANSPORT_RDMAOE_1_25	Added enumerated values
	is_roce(), is_mac_based_roce(), is_ip_based_roce(), is_roce_or_ip_based_roce()	Added new functions
	struct ib_wc_t	p_next was replaced with an anonymous union which contains two fields: p_next and qp_context
4.70	ib_get_port_spl_qp()	Added a new function
	ib_get_mad_inner()	Changed API (one more input parameter was added)
	ib_get_mad()	Changed API (one more input parameter was added)
	VERBS_MINOR_VER	Increased its value, 0x000a -> 0x000c
	UNBOUND_PORT_NUM	Added a new macro
4.60	IB_MOD_QP_CHANGE_COUNTER_IN DEX	Added a new macro
	struct ib_qp_mod_t	Added the field state.rtr.counter_index
4.55	RDMA_TRANSPORT_RRDMAOE_1_5	Added enumerated values
	RDMA_TRANSPORT_RRDMAOE_2_0	Added enumerated values

Release	Name	Description
4.50	is_roce(), is_xroce()	Added new functions
	IB_AC_SNIFFER	Added a new macro
	struct ib_qp_mod_t	Added the field state.init.flags
4.40	VERBS_MINOR_VER	Increased its value, 0x0009 -> 0x000a
	enum eth_link_speeds	Added enumerated values
	struct ib_port_attr_t	<ul style="list-style-type: none"> The mtu field was separated into two fields: <ul style="list-style-type: none"> max_mtu (maximum MTU supported by the port) active_mtu (actual MTU which the port is configured with) Added the eth_link_speed field
	WR_SEND_INV	Added enumerated values
	struct ib_send_wr_t	The type of invalidate_rkey was changed from net32_t -> ib_net32_t
	IB_SEND_OPT_SKIP_DOORBELL	Added the send Write flag