SOFTWARE MANUAL

Mellanox ConnectX-4/ConnectX-5 NATIVE ESXi Driver for VMware vSphere 6.5 User Manual

Rev 4.16.14.2

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

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

Table of Contents

List of Tables

Document Revision History

Table 1 - Document Revision History

Release	Date	Description
Rev 4.16.14.2	April 2, 2019	• Updated section Section 3.2.1, "Single Root IO Virtualization (SR-IOV)", on page 25, updated the supported VFs for ConnectX-4/ConnectX-5 adapter cards to 64/128 depending on the firmware capabilities.
Rev 4.16.13.5	October 22, 2018	• Added the following sections: • Section 3.1.6.5, "Explicit Congestion Notification (ECN)", on page 22 • Updated section Section 1.3.1.1, "nmlx5_core Parameters", on page 10
Rev 4.16.13.5	January 8, 2018	• Added the following sections: • Section 3.1.7, "Packet Capture Utility", on page 23 • Updated section Section 1.3.1.1, "nmlx5_core Parameters", on page 10

Rev 4.16.10.3	October 17, 2017	• Added the following sections:
		  • Section 3.1.1, "Port Type Management", on page 16
		  • Section 3.2.3, "Configuring InfiniBand-SR-IOV", on page 29
		  • Section 3.3, "Mellanox NIC ESXi Management Tools", on page 31
		  • Section 2.4, "Downgrading to an Older Mellanox Driver Version", on page 14

Rev 4.16.8.8	January 31, 2017	• Added section Section 3.2.4, "GENEVE Hardware Offload", on page 31 • Updated section Section 3.1.6, "RDMA over Converged Ethernet (RoCE)", on page 19
Rev 4.16.7.8	November 30, 2016	Initial version of this release.

About this Manual

This preface provides general information concerning the scope and organization of this User's Manual.

Intended Audience

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

Common Abbreviations and Acronyms

Table 2 - Abbreviations and Acronyms

Abbreviation / Acronym	Whole Word / Description
B	(Capital) 'B' is used to indicate size in bytes or multiples of bytes (e.g., 1KB = 1024 bytes, and 1MB = 1048576 bytes)
b	(Small) 'b' is used to indicate size in bits or multiples of bits (e.g., 1Kb = 1024 bits)
FW	Firmware
HCA	Host Channel Adapter
HW	Hardware
LSB	Least significant *byte*
lsb	Least significant *bit*
MSB	Most significant *byte*
msb	Most significant *bit*
NIC	Network Interface Card
SW	Software
VPI	Virtual Protocol Interconnect
PR	Path Record
RDS	Reliable Datagram Sockets
SDP	Sockets Direct Protocol
SL	Service Level
MPI	Message Passing Interface
QoS	Quality of Service
ULP	Upper Level Protocol
vHBA	Virtual SCSI Host Bus adapter
uDAPL	User Direct Access Programming Library

Related Documentation

Table 3 - Reference Documents

Document Name	Description
IEEE Std 802.3ae™-2002 (Amendment to IEEE Std 802.3-2002) Document # PDF: SS94996	Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications Amendment: Media Access Control (MAC) Parameters, Physical Layers, and Management Parameters for 10 Gb/s Operation
Firmware Release Notes for Mellanox adapter devices	See the Release Notes PDF file relevant to your adapter device. For further information please refer to the Mellanox website. www.mellanox.com -> Support -> Firmware Download
MFT User Manual	Mellanox Firmware Tools User's Manual. For further information please refer to the Mellanox website. www.mellanox.com -> Products -> Ethernet Drivers -> Firmware Tools
MFT Release Notes	Release Notes for the Mellanox Firmware Tools. For further information please refer to the Mellanox website. www.mellanox.com -> Products -> Ethernet Drivers -> Firmware Tools
VMware vSphere 6.5 Documentation Center	VMware website

1 Introduction

Mellanox ConnectX®-4/ConnectX-5 NATIVE ESXi is a software stack which operates across all Mellanox network adapter solutions supporting up to 100Gb/s Ethernet (ETH) and 2.5 or 5.0 GT/s PCI Express 2.0 and 3.0 uplinks to servers.

The following sub-sections briefly describe the various components of the Mellanox ConnectX-4/ConnectX-5 NATIVE ESXi stack.

1.1 nmlx5 Driver

`nmlx5` is the low level driver implementation for the ConnectX-4/ConnectX-5 adapter cards designed by Mellanox Technologies. ConnectX-4/ConnectX-5 adapter cards can operate as an InfiniBand adapter, or as an Ethernet NIC. The ConnectX-4/ConnectX-5 NATIVE ESXi driver supports Ethernet NIC configurations exclusively. In addition, the driver provides RDMA over Converged Ethernet (RoCE) functionality through ESXi RDMA layer APIs (kernel-space only) and SR-IOV.

1.2 Mellanox NATIVE ESXi Package

1.2.1 Software Components

MLNX-NATIVE-ESX-ConnectX-4/ConnectX-5 contains the following software components:

- Mellanox Host Channel Adapter Drivers
  - **nmlx5_core** (Ethernet): Handles Ethernet specific functions and plugs into the ESXi uplink layer
  - **nmlx5_rdma**: Enables RoCE functionality by plugging into the ESXi RDMA layer

1.3 Module Parameters

1.3.1 Module Parameters

To set **`nmlx5_core`** parameters:

```
esxcli system module parameters set -m nmlx5_core -p <parameter>=<value>
```

To set **`nmlx5`**`_rdma` parameters:

```
esxcli system module parameters set -m nmlx5_rdma -p <parameter>=<value>
```

To show the values of the parameters:

```
esxcli system module parameters list -m  <module name>
```

For the changes to take effect, reboot the host.

1.3.1.1 nmlx5_core Parameters

Table 1 - nmlx5_core Module Parameters

Name	Description	Values
DRSS	Number of hardware queues for Default Queue (DEFQ) RSS. **Note:** This parameter replaces the previously used "drss" parameter which is now obsolete.	• 2-16 • 0 - disabled When this value is != 0, DEFQ RSS is enabled with 1 RSS Uplink queue that manages the 'drss' hardware queues. **Notes:** • The value must be a power of 2. • The value must not exceed num. of CPU cores. • Setting the DRSS value to 16, sets the Steering Mode to device RSS
enable_nmlx_debug	Enables debug prints for the core module.	• 1 - enabled • 0 - disabled (Default)
geneve_offload_enable	Enables GENEVE HW Offload	• 1 - enabled • 0 - disabled (Default)
max_vfs	max_vfs is an array of comma separated integer values, that represent the amount of VFs to open from each port. For example: max_vfs = 1,1,2,2, will open a single VF per port on the first NIC and 2 VFs per port on second NIC. The order of the NICs is determined by pci SBDF number. **Note:** VFs creation based on the system resources limitations.	• 0 - disabled (Default) N number of VF to allocate over each port **Note:** The amount of values provided in the max_vfs array should not exceed the supported_num_ports module parameter value.
mst_recovery	Enables recovery mode (only NMST module is loaded).	• 1 - enabled • 0 - disabled (Default)
pfcrx	Priority based Flow Control policy on RX.	• 0-255 • 0 - default It is an 8 bits bit mask, where each bit indicates a priority [0-7]. Bit values: • 1 - respect incoming PFC pause frames for the specified priority. • 0 - ignore incoming pause frames on the specified priority. **Note**: The pfcrx and pfctx values must be identical.

Table 1 - nmlx5_core Module Parameters

Name	Description	Values
`pfctx`	Priority based Flow Control policy on TX.	• 0-255 • 0 - default It is an 8 bits bit mask, where each bit indicates a priority [0-7]. Bit values: • 1 - generate pause frames according to the RX buffer threshold on the specified priority. • 0 - never generate pause frames on the specified priority. **Note**: The pfcrx and pfctx values must be identical.
`RSS`	Number of hardware queues for NetQ RSS. **Note:** This parameter replaces the previously used "`rss`" parameter which is now obsolete.	• 2-8 • 0 - disabled When this value is != 0, NetQ RSS is enabled with 1 RSS uplink queue that manages the 'rss' hardware queues. Notes: • The value must be a power of 2 • The maximum value must be lower than the number of CPU cores.
`supported_num_ports`	Sets the maximum supported ports.	2-8 Default 4 **Note:** Before installing new cards, you must modify the maximum number of the supported ports to include the additional new ports.
`ecn`	Enables the ECN feature	• 1 - enable (default) • 0 - disabled

2 Installation

This chapter describes how to install and test the Mellanox ConnectX-4/ConnectX-5 NATIVE ESXi package on a single host machine with Mellanox Ethernet adapter hardware installed.

2.1 Hardware and Software Requirements

Table 2 - Software and Hardware Requirements

Requirements	Description
Platforms	A server platform with an adapter card based on one of the following Mellanox Technologies' HCA devices: • ConnectX®-4 (EN) (firmware: fw-ConnectX4) • ConnectX®-4 Lx (EN) (firmware: fw-ConnectX4-Lx) • ConnectX®-5 (VPI) (firmware: fw-ConnectX5) • ConnectX®-5 Ex (VPI) (firmware: fw-ConnectX5)
Device ID	For the latest list of device IDs, please visit Mellanox website.
Operating System	ESXi 6.5: 4.16.10.3
Installer Privileges	The installation requires administrator privileges on the target machine.

2.2 Installing Mellanox NATIVE ESXi Driver for VMware vSphere

Please uninstall any previous Mellanox driver packages prior to installing the new version. See Section 2.3, "Removing the Previous Mellanox Driver", on page 14 for further information.

➢ ***To install the driver:***

1. Download the appropriate driver version from the Mellanox site.
   The driver is downloaded from a dedicated link that points to the VMware site.
2. Unzip the downloaded archive/file.
   Note: The actual bundle is embedded in the downloaded zip from the VMware site, and not the zip itself.
3. Log into the ESXi server with root permissions.
4. Install the driver.

   ```
   #> esxcli software vib install -d <path>/<bundle_file>
   ```

   Example:

   ```
   #> esxcli software vib install -d /tmp/MLNX-NATIVE-ESX-ConnectX-4-5_4.16.10.3-10EM-650.0.0.2768847.zip
   ```

5. Reboot the machine.
6. Verify the driver was installed successfully.

   ```
   esxcli software vib list | grep nmlx
   nmlx5-core                4.16.10.3-1OEM.650.0.0.4598673        MEL      PartnerSupported  2017-01-31
   nmlx5-rdma                4.16.10.3-1OEM.650.0.0.4598673        MEL      PartnerSupported  2017-01-31
   ```

After the installation process, all kernel modules are loaded automatically upon boot.

2.3 Removing the Previous Mellanox Driver

Please unload the driver before removing it.

➢ ***To remove all the drivers:***

1. Log into the ESXi server with root permissions.
2. List all the existing NATIVE ESXi driver modules. (see Step 4 in Section 2.2, on page 13)
3. Remove each module.

```
#> esxcli software vib remove -n nmlx5-rdma
#> esxcli software vib remove -n nmlx5-core
```

To remove the modules, the command must be run in the same order as shown in the example above.

4. Reboot the server.

2.4 Downgrading to an Older Mellanox Driver Version

Please unload the driver before removing it.

➢ ***To downgrade to the previous ESXi version:***

Please note, automatic downgrade flow in currently unavailable for current driver version due to a change in the number of driver modules. Using "`esxcli software update`" command to downgrade may cause unexpected result.

In order to safely downgrade to any previous version (e.g., 4.16.8.8), you must **manually** remove the current version and install the previous one as described in the process below.

1. Log into the ESXi server with root permissions.
2. List all the existing NATIVE ESXi driver modules. (see Step 4 in Section 2.2, on page 13)

3. Remove each module.

```
#> esxcli software vib remove -n nmlx5-rdma
#> esxcli software vib remove -n nmlx5-core
```

To remove the modules, the command must be run in the same order as shown in the example above.

4. Install the desired driver version.
5. Reboot the machine.

2.5 Firmware Programming

1. Download the VMware bootable binary images v4.8.0 from the Mellanox Firmware Tools (MFT) site.
   - **ESXi 6.5 File:** mft-4.8.0.26-10EM-650.0.0.4598673.x86_64.vib

     MD5SUM: 6f4a1c1ef2482f091bee4086cbec5caf
2. Install the image according to the steps described in the MFT User Manual.

The following procedure requires custom boot image downloading, mounting and booting from a USB device.

3 Features Overview and Configuration

3.1 Ethernet Network

3.1.1 Port Type Management

ConnectX®-4/ConnectX®-4 Lx/ConnectX®-5 ports can be individually configured to work as InfiniBand or Ethernet ports. The port type depends on the card type. In case of a VPI card, the default type is IB. If you wish to change the port type use the `mlxconfig` script.

To use a VPI card as an Ethernet only card, run:

```
/opt/mellanox/bin/mlxconfig -d /dev/mt4115_pciconf0 set LINK_TYPE_P1=2 LINK_TYPE_P2=2
```

The protocol types are:

- Port Type 1 = IB
- Port Type 2 = Ethernet

For further information on how to set the port type in ConnectX®-4/ConnectX®-4 Lx/ConnectX®-5, please refer to the MFT User Manual (www.mellanox.con --> Products --> Software --> InfiniBand/VPI Software --> MFT - Firmware Tools).

3.1.2 Wake-on-LAN (WoL)

Please note that Wake-on-LAN (WOL) is applicable only to adapter cards that support this feature.

Wake-on-LAN (WOL) is a technology that allows a network professional to remotely power on a computer or to wake it up from sleep mode.

- To enable WoL:

  ```
  esxcli network nic set -n <nic name> -w g
  ```

  or

  ```
  set /net/pNics/<nic name>/wol g
  ```

- To disable WoL:

  ```
  vsish -e set /net/pNics/<nic name>/wol d
  ```

- To verify configuration:

```
esxcli network nic get -n vmnic5
   Advertised Auto Negotiation: true
   Advertised Link Modes: 10000baseT/Full, 40000baseT/Full, 100000baseT/Full, 100baseT/
Full, 1000baseT/Full, 25000baseT/Full, 50000baseT/Full
   Auto Negotiation: false
   Cable Type: DA
   Current Message Level: -1
   Driver Info:
         Bus Info: 0000:82:00:1
         Driver: nmlx5_core
         Firmware Version: 12.20.1010
         Version: 4.15.10.3
   Link Detected: true
   Link Status: Up
   Name: vmnic5
   PHYAddress: 0
   Pause Autonegotiate: false
   Pause RX: false
   Pause TX: false
   Supported Ports:
   Supports Auto Negotiation: true
   Supports Pause: false
   Supports Wakeon: false
   Transceiver:
   Wakeon: MagicPacket(tm)
```

3.1.3 Set Link Speed

The driver is set to auto-negotiate by default. However, the link speed can be forced to a specific link speed supported by ESXi using the following command:

```
esxcli network nic set -n <vmnic> -S <speed> -D <full, half>
```

Example:

```
esxcli network nic set -n vmnic4 -S 10000 -D full
```

where:

- `<speed>` in ESXi 6.5 can be 10/100/1000/2500/5000/10000/20000/25000/40000/50000/56000/100000Mb/s.
- `<vmnic>` is the vmnic for the Mellanox card as provided by ESXi
- `<full, half>` The duplex to set this NIC to. Acceptable values are: [full, half]

The driver can be reset to auto-negotiate using the following command:

```
esxcli network nic set -n <vmnic> -a
```

Example:

```
esxcli network nic set -n vmnic4 -a
```

where `<vmnic>` is the vmnic for the Mellanox card as provided by ESXi.

3.1.4 Priority Flow Control (PFC)

Priority Flow Control (PFC) IEEE 802.1Qbb applies pause functionality to specific classes of traffic on the Ethernet link. PFC can provide different levels of service to specific classes of Ethernet traffic (using IEEE 802.1p traffic classes).

When PFC is enabled, Global Pause will be operationally disabled, regardless of what is configured for the Global Pause Flow Control.

➢ ***To configure PFC:***

Step 1. Enable PFC for specific priorities.

```
esxcfg-module nmlx5_core -s "pfctx=0x08 pfcrx=0x08"
```

The parameters, "pfctx" (PFC TX) and "pfcrx" (PFC RX), are specified per host. If you have more than a single card on the server, all ports will be enabled with PFC (Global Pause will be disabled even if configured).

The value is a bitmap of 8 bits = 8 priorities. We recommend that you enable only lossless applications on a specific priority.

To run more than one flow type on the server, turn on only one priority (e.g. priority 3), which should be configured with the parameters "0x08" = 00001000b (binary). Only the 4th bit is on (starts with priority 0,1,2 and 3 -> 4th bit).

Note: The values of "pfctx" and "pfcrx" must be identical.

Step 2. Restart the driver.

```
reboot
```

3.1.5 Receive Side Scaling (RSS)

Receive Side Scaling (RSS) technology allows spreading incoming traffic between different receive descriptor queues. Assigning each queue to different CPU cores allows better load balancing of the incoming traffic and improve performance.

3.1.5.1 Default Queue Receive Side Scaling (DRSS)

Default Queue RSS (DRSS) allows the user to configure multiple hardware queues backing up the default RX queue. DRSS improves performance for large scale multicast traffic between hypervisors and Virtual Machines interfaces.

To configure DRSS, use the `'DRSS'` module parameter which replaces the previously advertised `'device_rss'` module parameter (`'device_rss'` is now obsolete). The `'drss'` module parameter and `'device_rss'` are mutually exclusive

If the `'device_rss'` module parameter is enabled, the following functionality will be configured:

- The new Default Queue RSS mode will be triggered and all hardware RX rings will be utilized, similar to the previous `'device_rss'` functionality

- Module parameters 'DRSS' and 'RSS' will be ignored, thus the NetQ RSS, or the standard NetQ will be active

To query the 'DRSS' module parameter default, its minimal or maximal values, and restrictions, run a standard esxcli command.

For example:

```
#esxcli system module parameters list -m nmlx5_core
```

3.1.5.2 NetQ RSS

NetQ RSS is a new module parameter for ConnectX-4 adapter cards providing identical functionality as the ConnectX-3 module parameter 'num_rings_per_rss_queue'. The new module parameter allows the user to configure multiple hardware queues backing up the single RX queue. NetQ RSS improves vMotion performance and multiple streams of IPv4/IPv6 TCP/UDP/IPSEC bandwidth over single interface between the Virtual Machines.

To configure NetQ RSS, use the 'RSS' module parameter. To query the 'RSS' module parameter default, its minimal or maximal values, and restrictions, run a standard esxcli command.

For example:

```
#esxcli system module parameters list -m nmlx5_core
```

Using NetQ RSS is preferred over the Default Queue RSS. Therefore, if both module parameters are set but the system lacks resources to support both, NetQ RSS will be used instead of DRSS.

3.1.5.3 Important Notes

If the 'DRSS' and 'RSS' module parameters set by the user cannot be enforced by the system due to lack of resources, the following actions are taken in a sequential order:

1. The system will attempt to provide the module parameters default values instead of the ones set by the user
2. The system will attempt to provide 'RSS' (NetQ RSS mode) default value. The Default Queue RSS will be disabled
3. The system will load with only standard NetQ queues
4. 'DRSS' and 'RSS' parameters are disabled by default, and the system loads with standard NetQ mode

3.1.6 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 lossless 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 and 40GigE 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.

When working with RDMA applications over Ethernet link layer the following points should be noted:

- The presence of a Subnet Manager (SM) is not required in the fabric. Thus, operations that require communication with the SM are managed in a different way in RoCE. This does not affect the API but only the actions such as joining multicast group, that need to be taken when using the API
- Since LID is a layer 2 attribute of the InfiniBand protocol stack, it is not set for a port and is displayed as zero when querying the port
- With RoCE, the alternate path is not set for RC QP and therefore APM is not supported
- GID format can be of 2 types, IPv4 and IPv6. IPv4 GID is a IPv4-mapped IPv6 address[1] while IPv6 GID is the IPv6 address itself
- VLAN tagged Ethernet frames carry a 3-bit priority field. The value of this field is derived from the InfiniBand SL field by taking the 3 least significant bits of the SL field
- RoCE traffic is not shown in the associated Ethernet device's counters since it is off-loaded by the hardware and does not go through Ethernet network driver. RoCE traffic is counted in the same place where InfiniBand traffic is counted:

```
esxcli rdma device stats get -d [RDMA device]
```

It is recommended to use RoCE with PFC enabled in driver and network switches.

For how to enable PFC in the driver see section Section 3.1.4, “Priority Flow Control (PFC)”, on page 18

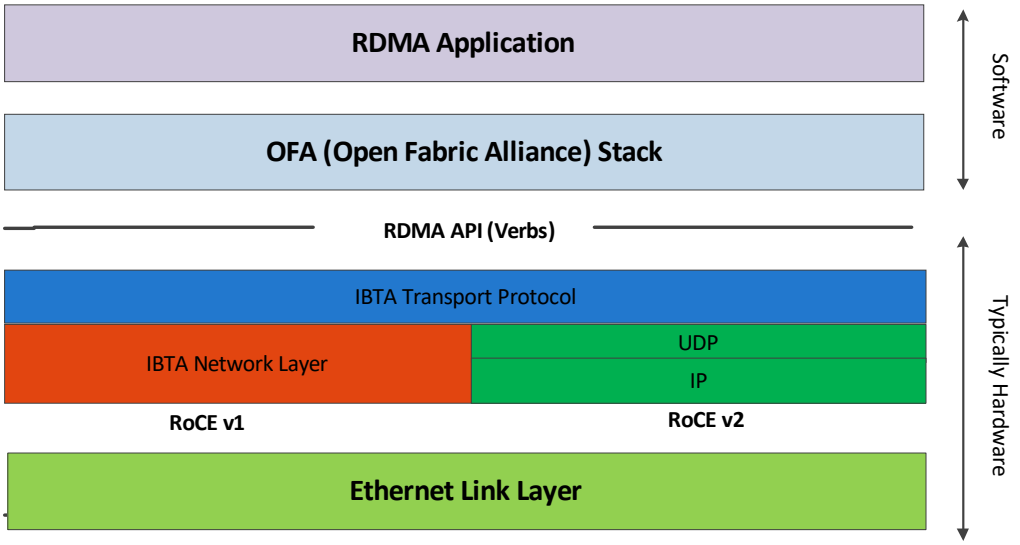
1. For the IPv4 address A.B.C.D the corresponding IPv4-mapped IPv6 address is ::ffff.A.B.C.D

3.1.6.1 RoCE Modes

RoCE encapsulates InfiniBand transport in one of the following Ethernet packet

- RoCEv1 - dedicated ether type (0x8915)
- RoCEv2 - UDP and dedicated UDP port (4791)

Figure 1: RoCEv1 and RoCEv2 Protocol Stack



3.1.6.1.1 RoCEv1

RoCE v1 protocol is defined as RDMA over Ethernet header (as shown in the figure above). It uses ethertype 0x8915 and may can be used with or without the VLAN tag. The regular Ethernet MTU applies on the RoCE frame.

3.1.6.1.2 RoCEv2

A straightforward extension of the RoCE protocol enables traffic to operate in IP layer 3 environments. This capability is obtained via a simple modification of the RoCE packet format. Instead of the GRH used in RoCE, IP routable RoCE packets carry an IP header which allows traversal of IP L3 Routers and a UDP header (RoCEv2 only) that serves as a stateless encapsulation layer for the RDMA Transport Protocol Packets over IP.

The proposed RoCEv2 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.

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, in a completely transparent way.

3.1.6.2 GID Table Population

The GID table is automatically populated by the ESXi RDMA stack using the 'binds' mechanism, and has a maximum size of 128 entries per port. Each bind can be of type RoCE v1 or RoCE v2, where entries of both types can coexist on the same table. Binds are created using IP-based GID generation scheme.

For more information, please refer to the "VMkernel APIs Reference Manual".

3.1.6.3 Prerequisites

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

- ConnectX®-4 firmware version 12.17.2020 and above
- ConnectX®-4 Lx firmware version 14.17.2020 and above
- ConnectX®-5 firmware version 16.20.1000 and above
- All InfiniBand verbs applications which run over InfiniBand verbs should work on RoCE links if they use GRH headers.
- All ports must be set to use Ethernet protocol

3.1.6.4 Running and Configuring RoCE on ESXi VMs

RoCE on ESXi VMs can run on VMs which are associated with either SR-IOV EN Virtual Functions or passthrough.

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.

On ESXi, the PFC settings should be set on the ESXi host only and not on the VMs as the ESXi host is the one to control PFC settings. PFC settings can be changed using the mlx5_core parameters `pfctx` and `pfcrx`. For further information, please refer to Section 1.3.1.1, "nmlx5_core Parameters", on page 10.

For further information on how to use and run RoCE on the VM, please refer t the VM's driver User Manual. Additional information can be found at the *RoCE Over L2 Network Enabled with PFC* User Guide:

http://www.mellanox.com/related-docs/prod_software/RoCE_with_Priority_Flow_Control_Application_Guide.pdf

3.1.6.5 Explicit Congestion Notification (ECN)

Explicit Congestion Notification (ECN) is an extension to the Internet Protocol and to the Transmission Control Protocol and is defined in RFC 3168 (2001). ECN allows end-to-end notification of network congestion without dropping packets. ECN is an optional feature that may be used between two ECN-enabled endpoints when the underlying network infrastructure also supports it.

ECN is enabled by default (ecn=1). To disable it, set the "`ecn`" module parameter to 0. For most use cases, the default setting of the ECN are sufficient. However, if further changes are required, use the nmlxcli management tool to tune the ECN algorithm behavior. For further information on

the tool, see Section 3.3, “Mellanox NIC ESXi Management Tools”, on page 31. The nmlxcli management tool can also be used to provide ECN different statistics.

3.1.7 Packet Capture Utility

Packet Capture utility duplicates all traffic, including RoCE, in its raw Ethernet form (before stripping) to a dedicated "sniffing" QP, and then passes it to an ESX drop capture point.

It allows gathering of Ethernet and RoCE bidirectional traffic via pktcap-uw and viewing it using regular Ethernet tools, e.g. Wireshark.

By nature, RoCE traffic is much faster than ETH. Meaning there is a significant gap between RDMA traffic rate and Capture rate.

Therefore actual "sniffing" RoCE traffic with ETH capture utility is not feasible for long periods.

3.1.7.1 Components

Packet Capture Utility is comprised of two components:

- ConnectX-4 RDMA module sniffer:

  This component is part of the Native ConnectX-4 RDMA driver for ESX and resides in Kernel space.

- RDMA management interface:

  User space utility which manages the ConnectX-4 Packet Capture Utility

3.1.7.2 Usage

Step 1. Installed the latest ConnectX-4 driver bundle.

Step 2. Make sure all Native nmlx5 drivers are loaded.

```
esxcli system module list | grep nmlx
nmlx5_core                     true        true
nmlx5_rdma                     true        true
```

Step 3. Install the nmlxcli management tool (esxcli extension) using the supplied bundle MLNX-NATIVE-NMLXCLI_1.16.12.11-10EM-650.0.0.4598673.zip

```
esxcli software vib install -d <path to bundle>/MLNX-NATIVE-NMLXCLI_1.16.12.11-10EM-650.0.0.4598673.zip
```

When the nmlxcli management tool is installed, the following esxli commands namespace is available:

```
# esxcli mellanox uplink sniffer
```

This namespace allows user basic packet capture utility operations such as: query, enable or disable.

Usage of the tool is shown by running one of the options below:

```
esxcli mellanox uplink sniffer {cmd} [cmd options]
```

Options:

```
disable      Disable sniffer on specified uplink
             * Requires -u/--uplink-name parameter
enable       Enable sniffer on specified uplink
             * Requires -u/--uplink-name parameter
query        Query operational state of sniffer on specified uplink
             * Requires -u/--uplink-name parameter
```

Step 4. Determine the uplink device name.

```
Name    PCI Device    Driver      Admin Status  Link Status   Speed  Duplex  MAC
Address         MTU  Description
------  ------------  ----------  ------------  -----------  ------  ------  -----
------------  ----  ----------------------------------------------------------

vmnic4  0000:07:00.0  nmlx5_core  Up            Up           100000  Full
7c:fe:90:63:f2:d6  1500  Mellanox Technologies MT27700 Family [ConnectX-4]
vmnic5  0000:07:00.1  nmlx5_core  Up            Up           100000  Full
7c:fe:90:63:f2:d7  1500  Mellanox Technologies MT27700 Family [ConnectX-4]
```

Step 5. Enable the packet capture utility for the required device(s).

```
esxcli mellanox uplink sniffer enable -u <vmnic_name>
```

Step 6. Use the ESX internal packet capture utility to capture the packets.

```
pktcap-uw --capture Drop --o <capture_file>
```

Step 7. Generate the RDMA traffic through the RDMA device.

Step 8. Stop the capture.

Step 9. Disable the packet capture utility.

```
esxcli mellanox uplink sniffer disable -u <vmnic_name>
```

Step 10. Query the packet capture utility.

```
esxcli mellanox uplink sniffer query -u <vmnic_name>
```

3.1.7.3 Limitations

- **Capture duration:** Packet Capture Utility is a debug tool, meant to be used for bind failure diagnostics and short period packet sniffing. Running it for a long period of time with stress RDMA traffic will cause undefined behavior. Gaps in capture packets may appear.
- **Overhead:** A significant performance decrease is expected when the tool is enabled:
  - The tool creates a dedicated QP and HW duplicates all RDMA traffic to this QP, before stripping the ETH headers.
  - The captured packets reported to ESX are duplicated by the network stack adding to the overhaul execution time
- **Drop capture point:** The tool uses the VMK_PKTCAP_POINT_DROP to pass the captured traffic. Meaning whomever is viewing the captured file will see all RDMA capture in addition to all the dropped packets reported to the network stack.

- **ESX packet exhaustion:** During the enable phase (/opt/mellanox/bin/nmlx4_sniffer_mgmt-user -a vmrdma3 -e) the Kernel component allocates sniffer resources, and among these are the OS packets which are freed upon tool's disable. Multiple consecutive enable/disable calls may cause temporary failures when the tool requests to allocate these packets. It is recommended to allow sufficient time between consecutive disable and enable to fix this issue.

3.2 Virtualization

3.2.1 Single Root IO Virtualization (SR-IOV)

Single Root IO 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-4/ConnectX-5 adapter cards up to 64/128 virtual instances called Virtual Functions (VFs), depending on the firmware capabilities. These virtual functions can then be provisioned separately. Each VF can be seen as an addition device connected to the Physical Function. It shares the same 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.

In this chapter we will demonstrate setup and configuration of SR-IOV in a ESXi environment using Mellanox ConnectX® adapter cards family.

3.2.1.1 System Requirements

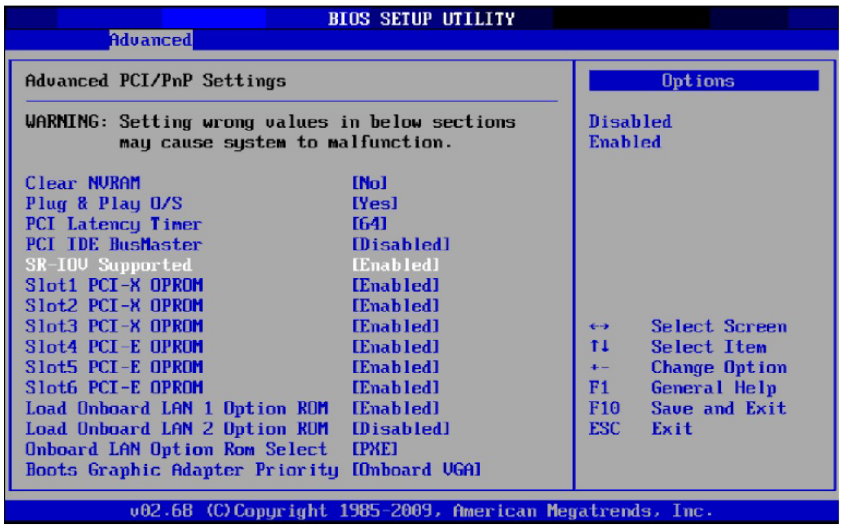
To set up an SR-IOV environment, the following is required:

- nmlx5_core Driver
- A server/blade with an SR-IOV-capable motherboard BIOS
- Mellanox ConnectX® Adapter Card family with SR-IOV capability
- Hypervisor that supports SR-IOV such as: ESXi 6.5

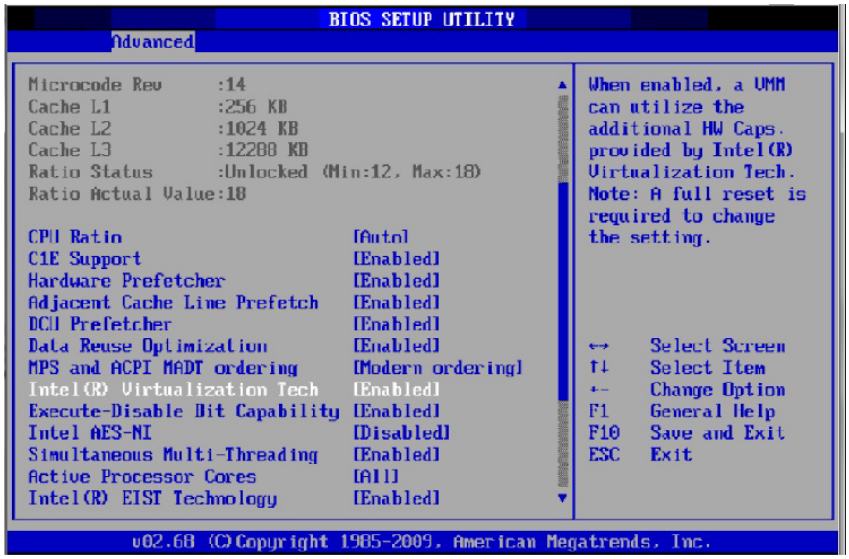
3.2.1.2 Setting Up SR-IOV

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:

Step 1. Enable "SR-IOV" in the system BIOS.



Step 2. Enable "Intel Virtualization Technology".



Step 3. Install ESXi 6.5 that support SR-IOV.

3.2.1.2.1 Configuring SR-IOV for ConnectX-4/ConnectX-5

Step 1. Install the MLNX-NATIVE-ESX-ConnectX-4/ConnectX-5 driver for ESXi that supports SR-IOV.

Step 2. Download the MFT package. Go to:
www.mellanox.com --> Products --> Software --> InfiniBand/VPI Drivers --> MFT (http://www.mellanox.com/page/management_tools)

Step 3. Install MFT.

```
# esxcli software vib install -v <MST Vib>
# esxcli software vib install -v <MFT Vib>
```

Step 4. Reboot system.

Step 5. Start the mst driver.

```
# /opt/mellanox/bin/mst start
```

Step 6. Check if SR-IOV is enabled in the firmware.

```
/opt/mellanox/bin/mlxconfig -d /dev/mst/mt4115_pciconf0 q

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

  Device type:    ConnectX4
  PCI device:     /dev/mst/mt4115_pciconf0
  Configurations:          Current
     SRIOV_EN              1
     NUM_OF_VFS            8
     FPP_EN                1
```

If not, use mlxconfig to enable it.

```
mlxconfig -d /dev/mst/mt4115_pciconf0 set SRIOV_EN=1 NUM_OF_VFS=16
```

Step 7. Power cycle the server.

Step 8. Set the number of Virtual Functions you need to create for the PF using the `max_vfs` module parameter.

```
esxcli system module parameters set -m nmlx5_core -p "max_vfs=8"
```

Note: The number of `max_vf` is set per port. See Table 1, “nmlx5_core Module Parameters,” on page 10 for further information.

3.2.1.3 Assigning a Virtual Function to a Virtual Machine in the vSphere Web Client

After you enable the Virtual Functions on the host, each of them becomes available as a PCI device.

➢ ***To assign Virtual Function to a Virtual Machine in the vSphere Web Client:***

Step 1. Locate the Virtual Machine in the vSphere Web Client.

a.Select a data center, folder, cluster, resource pool, or host and click the Related Objects tab.

b. Click Virtual Machines and select the virtual machine from the list.

Step 2. Power off the Virtual Machine.

Step 3. On the **Manage** tab of the Virtual Machine, select **Settings > VM Hardware**.

Step 4. Click **Edit** and choose the **Virtual Hardware** tab.

Step 5. From the **New Device** drop-down menu, select **Network** and click **Add**.

Step 6. Expand the **New Network** section and connect the Virtual Machine to a port group.

The virtual NIC does not use this port group for data traffic. The port group is used to extract the networking properties, for example VLAN tagging, to apply on the data traffic.

Step 7. From the **Adapter Type** drop-down menu, select **SR-IOV passthrough**.

Step 8. From the **Physical Function** drop-down menu, select the **Physical Adapter** to back the passthrough Virtual Machine adapter.

Step 9. **[Optional]** From the **MAC Address** drop-down menu, select **Manual** and type the static MAC address.

Step 10. Use the **Guest OS MTU Change** drop-down menu to allow changes in the MTU of packets from the guest operating system.
Note: This step is applicable only if this feature is supported by the driver.

Step 11. Expand the **Memory** section, select **Reserve all guest memory (All locked)** and click **OK**.

I/O memory management unit (IOMMU) must reach all Virtual Machine memory so that the passthrough device can access the memory by using direct memory access (DMA).

Step 12. Power on the Virtual Machine.

3.2.2 VXLAN Hardware Offload

VXLAN hardware offload enables the traditional offloads to be performed on the encapsulated traffic. With ConnectX® family adapter cards, data center operators can decouple the overlay network layer from the physical NIC performance, thus achieving native performance in the new network architecture.

3.2.2.1 Configuring VXLAN Hardware Offload

VXLAN hardware offload includes:

- TX: Calculates the Inner L3/L4 and the Outer L3 checksum
- RX:
  - Checks the Inner L3/L4 and the Outer L3 checksum
  - Maps the VXLAN traffic to an RX queue according to:
    - Inner destination MAC address
    - Outer destination MAC address
    - VXLAN ID

VXLAN hardware offload is enabled by default and its status cannot changed.

VXLAN configuration is done in the ESXi environment via VMware NSX manager. For additional NSX information, please refer to VMware documentation, see:
http://pubs.vmware.com/NSX-62/index.jsp#com.vmware.nsx.install.doc/GUID-D8578F6E-A40C-493A-9B43-877C2B75ED52.html.

3.2.3 Configuring InfiniBand-SR-IOV

InfiniBand SR-IOV is tested only on Windows Server 2016.

Step 1. Install nmlx5 driver version 4.16.10-3 or above.

Step 2. Install MFT version 4.7.0-42 or above.

```
# esxcli software vib install -d MLNX-NMFT-ESX_4.7.0.42-10EM-650.0.0.4598673.zip
# reboot
```

Step 3. Query the firmware configuration to locate the device.

```
# cd /opt/mellanox/bin
# ./mlxconfig q
Device type:    ConnectX4
PCI device:     mt4115_pciconf0
```

Step 4. Use MFT to burn the latest firmware version.

```
# flint -d mt4115_pciconf0 b -i fw-ConnectX4-rel-12_20_1010-MCX456A-ECA_Ax-Flex-
Boot-3.5.210.bin b
# reboot
```

Step 5. Set the link type of one or both ports to InfiniBand.

```
# cd /opt/mellanox/bin
# ./mlxconfig -d mt4115_pciconf0 set LINK_TYPE_P1=1 (LINK_TYPE_P2=1)
```

One InfiniBand port per subnet must be dedicated to running the Subnet Manager (SM). Since the SM can only run on PFs, that port must be passthroughed to a VM.

Step 6. Enable Ethernet PCI subclass override.

```
# ./mlxconfig -d mt4115_pciconf0 set ADVANCED_PCI_SETTINGS=1
# ./mlxconfig -d mt4115_pciconf0 set FORCE_ETH_PCI_SUBCLASS=1
```

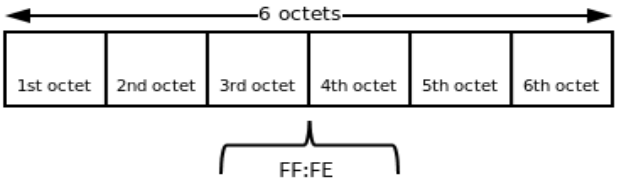
Step 7. Set the "max_vfs" module parameter to the preferred number of VFs.

```
# esxcfg-module nmlx5_core -s "max_vfs=2"
# reboot
```

InfiniBand ports are displayed on the ESXi host as “downed” uplinks and have no data path. The data path exists only for Guest OS.

Step 8. Assign the InfiniBand SR-IOV VFs to the VMs.For further information on how to assign the VFs, see Section 3.2.1.3, “Assigning a Virtual Function to a Virtual Machine in the vSphere Web Client”, on page 27

When ESXi sets the MAC for an InfiniBand VF, the formula used to convert it to GUID is adding "FF:FE" between the 3rd and the 4th octet:



For example:

```
12:34:56:78:9A:BC --> 12:34:56:FF:FE:78:9A:BC
```

When assigning VFs in InfiniBand SR-IOV, the value set for MTU is ignored.

Step 9. Configure the Subnet Manager.

Step a. Passthrough an InfiniBand PF to a VM.

Step b. Create an OpenSM config file

```
opensm --create-config /etc/opensm.conf
```

Step c. Add to the opensm.conf file "`virt_enabled 2`".

Step d. Run OpenSM.

```
opensm --config /etc/opensm.conf
```

If configured correctly, the link state of the VFs should be “Active”.

Please refer to the Mellanox OFED User Manual for further information.

http://www.mellanox.com/related-docs/prod_software/Mellanox_OFED_Linux_User_Manual_v4.1.pdf

Do not forget to enable virtualization in the Subnet Manager configuration (see section "Configuring SR-IOV for ConnectX-4/Connect-IB (InfiniBand) "Step 7" in Mellanox OFED User Manual).

Communication of InfiniBand VFs is GID based only, and requires every message to include GRH header. Thus, when using `ib_write_*/ib_send_*` tools, "`-x 0`" option must be specified explicitly.

3.2.4 GENEVE Hardware Offload

GENEVE hardware offload enables the traditional offloads to be performed on the encapsulated traffic. With ConnectX-4/ConnectX-5 family adapter cards, data center operators can decouple the overlay network layer from the physical NIC performance, thus achieving native performance in the new network architecture.

3.2.4.1 Configuring GENEVE Hardware Offload

GENEVE hardware offload includes:

- TX: Calculates the Inner L3/L4 and the Outer L3 checksum
- RX:
  - Checks the Inner L3/L4 and the Outer L3 checksum
  - Maps the GENEVE traffic to an RX queue according to:
    - Inner destination MAC address
    - Outer destination MAC address
    - GENEVE VNI

GENEVE hardware offload is disabled by default.

To enable it, run the "`geneve_offload_enable`" parameter.

GENEVE configuration is done in the ESXi environment via VMware NSX manager. For additional NSX information, please refer to VMware documentation, see:
http://pubs.vmware.com/NSX-62/index.jsp#com.vmware.nsx.install.doc/GUID-D8578F6E-A40C-493A-9B43-877C2B75ED52.html.

3.3 Mellanox NIC ESXi Management Tools

nmlxcli tools is a Mellanox esxcli command line extension for ConnectX®-3/ConnectX®-4/ ConnectX®-5 drivers' management for ESXi 6.0 and later.

This tool enables querying of Mellanox NIC and driver properties directly from driver / firmware.

Once the tool bundle is installed (see Section 3.3.2, "Installing nmlxcli", on page 32), a new NameSpace named `'mellanox'` will be available when executing main #esxcli command, containing additional nested NameSpaces and available commands for each NameSpace.

For general information on 'esxcli' commands usage, syntax, NameSpaces and commands, refer to the VMware vSphere Documentation Center:

https://pubs.vmware.com/vsphere-65/topic/com.vmware.vcli.getstart.doc/GUID-CDD49A32-91DB-454D-8603-3A3E4A09DC59.html

During `'nmlxcli'` commands execution, most of the output is formatted using the standard esxcli formatter, thus if required, the option of overriding the standard formatter used for a given command is available, for example:

Executing `'esxcli --formatter=xml mellanox uplink list'` produces XML output of given command.

For general information on esxcli generated output formatter, refer to the VMware vSphere Documentation Center:

https://pubs.vmware.com/vsphere-65/topic/com.vmware.vcli.examples.doc/GUID-227F889B-3EC0-48F2-85F5-BF5BD3946AA9.html

The current implementation does not support private statistics output formating.

In case of execution failure, the utility will prompt to standard output or/and log located at `'/var/log/syslog.log'`.

3.3.1 Requirements

Mellanox 'nmlxcli' tool is compatible with:

- ConnectX-3 driver version 3.15.10.3 and above
- ConnectX-4/5 driver version 4.16.10.3 and above

3.3.2 Installing nmlxcli

nmlxcli installation is performed as standard offline bundle.

➢ ***To install nmlxcli:***

Step 1. Run

```
esxcli software vib install -d <path_to_nmlxcli_extension_bundle.zip>
```

For general information on updating ESXi from a zip bundle, refer to the VMware vSphere Documentation Center:

https://pubs.vmware.com/vsphere-65/topic/com.vmware.vsphere.upgrade.doc/GUID-22A4B153-CB21-47B4-974E-2E5BB8AC6874.html

Step 2. For the new Mellanox namespace to function:

- Restart the ESXi host daemon.

  ```
  /etc/init.d/hostd restart
  ```

  or

- reboot ESXi host.