



Connect. Accelerate. Outperform.™

Mellanox Windows Network Adapter Management (CIM) for Windows Platforms

Rev 4.80

www.mellanox.com

NOTE:

THIS HARDWARE, SOFTWARE OR TEST SUITE PRODUCT (“PRODUCT(S)”) AND ITS RELATED DOCUMENTATION ARE PROVIDED BY MELLANOX TECHNOLOGIES “AS-IS” WITH ALL FAULTS OF ANY KIND AND SOLELY FOR THE PURPOSE OF AIDING THE CUSTOMER IN TESTING APPLICATIONS THAT USE THE PRODUCTS IN DESIGNATED SOLUTIONS. THE CUSTOMER'S MANUFACTURING TEST ENVIRONMENT HAS NOT MET THE STANDARDS SET BY MELLANOX TECHNOLOGIES TO FULLY QUALIFY THE 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

Mellanox Technologies, Ltd.
Beit Mellanox
PO Box 586 Yokneam 20692
Israel
www.mellanox.com
Tel: +972 (0)74 723 7200
Fax: +972 (0)4 959 3245

© Copyright 2014. Mellanox Technologies. All Rights Reserved.

Mellanox®, Mellanox logo, BridgeX®, ConnectX®, Connect-IB®, CORE-Direct®, InfiniBridge®, InfiniHost®, InfiniScale®, MetroX®, MLNX-OS®, PhyX®, ScalableHPC®, SwitchX®, UFM®, Virtual Protocol Interconnect® and Voltaire® are registered trademarks of Mellanox Technologies, Ltd.

ExtendX™, FabricIT™, Mellanox Open Ethernet™, Mellanox Virtual Modular Switch™, MetroDX™, TestX™, Unbreakable-Link™ are trademarks of Mellanox Technologies, Ltd.

All other trademarks are property of their respective owners.

Contents

About this Manual	9
1 Overview	10
1.1 CIM Management Flow	10
1.2 General Design and Implementation Guidelines	10
2 CIM Support Requirements and Settings	12
2.1 Supported CIM Profiles	12
2.2 Supported CIM Operations	12
2.3 CIM Object Manager (CIMOM) Requirements	12
2.4 Supported Operating Systems	12
2.5 Configuring CIM Provider	13
2.5.1 Remote Execution	13
2.5.2 Software Installation	13
3 Data Model	14
3.1 Device Level View	14
3.2 Network Adapter (Port) Level View	14
3.3 Software Inventory View	16
3.4 Software Update View	17
3.5 Common Diagnostic Model (CDM) View	18
3.6 Driver Level Management View	18
3.7 CIM Indications View	20
4 Using CIM Features	21
4.1 Overview	21
4.2 Using CIM cmdlets	21
4.3 Namespace	22
4.4 Establishing the Session	22
4.5 Get Instance	23
4.6 Invoke Method	23
4.7 Association	24
4.8 Events' Registry	24
4.9 Custom Cmdlets	24
5 Profile Classes, Methods and Properties	25
5.1 Profile Registration	25
5.1.1 MLNX_RegisteredProfile – Extend CIM_RegisteredProfile	25
5.1.2 MLNX_ElementConformsToProfile – Extend CIM_ElementConformsToProfile	25
5.2 Software Inventory Profile	26
5.2.1 MLNX_SoftwareIdentity – Extend CIM_SoftwareIdentity	26

- 5.2.2 MLNX_ FirmwareIdentity – Extend CIM_SoftwareIdentity 27
- 5.2.3 MLNX_ DriverIdentity – Extend CIM_SoftwareIdentity..... 28
- 5.2.4 MLNX_ ElementSoftwareIdentity – Extend CIM_ElementSoftwareIdentity 28
- 5.2.5 MLNX_ InstalledSoftwareIdentity – Extend CIM_InstalledSoftwareIdentity 29
- 5.3 Software Update Profile..... 29
 - 5.3.1 MLNX_ SoftwareInstallationService – Extend CIM_SoftwareInstallationService.... 29
 - 5.3.2 MLNX_ FirmwareInstallationService – Extend CIM_SoftwareInstallationService ... 31
- 5.4 PCI Device Profile..... 32
 - 5.4.1 MLNX_PCIDevice – Extend CIM_PCIDevice 32
 - 5.4.2 MLNX_SystemDevice – Extend CIM_SystemDevice..... 34
 - 5.4.3 MLNX_PCIDeviceCapabilities – Extend CIM_Capabilities 34
 - 5.4.4 MLNX_PCIDeviceSettingData – Extend CIM_SettingData..... 35
 - 5.4.5 MLNX_PCIDeviceSriovSettingData – Extend MLNX_PCIDeviceSettingData 35
 - 5.4.6 MLNX_PCIDevicePortTypeSettingData – Extend MLNX_PCIDeviceSettingData. 36
 - 5.4.7 MLNX_IBPort – Extend CIM_ IBPort..... 37
 - 5.4.8 MLNX_ IBPortCounters – Extend CIM_IBPortStatistics..... 38
 - 5.4.9 MLNX_IBPortControlledBy – Extend CIM_ ControlledBy 39
- 5.5 Ethernet Port Profile 39
 - 5.5.1 MLNX_NetAdapter – Extend CIM_ EthernetPort 39
 - 5.5.2 MLNX_NetAdapterStatistics – Extend CIM_NetworkPortStatistics..... 40
 - 5.5.3 MLNX_NetAdapterControlledBy – Extend CIM_ ControlledBy 42
 - 5.5.4 MLNX_NetAdapterFlowControlSettingData extend CIM_SettingData..... 42
- 5.6 Physical Asset Profile 43
 - 5.6.1 MLNX_Card – Extend CIM_Card 43
 - 5.6.2 MLNX_Realizes – Extend CIM_ Realizes 44
- 5.7 Host LAN Network Port Profile 44
- 5.8 Diagnostics CDMv2 Profile..... 44
 - 5.8.1 MLNX_DiagnosticTestHca – Extend MLNX_DiagnosticTest 45
 - 5.8.2 MLNX_DiagnosticConcreteJob – Extend CIM_ConcreteJob 46
 - 5.8.3 MLNX_DiagnosticLog – Extend CIM_DiagnosticJob 47
 - 5.8.4 MLNX_AvailableDiagnosticService – Extend CIM_ AvailableDiagnosticService... 47
 - 5.8.5 MLNX_DiagnosticUseOfLog – Extend CIM_ UseOfLog 47
- 5.9 Indications Profile 48
- 5.10 Drivers Services Profile 48
 - 5.10.1 MLNX_DriverService – Extend CIM_Service 48
 - 5.10.2 MLNX_DriverCoreCapabilities – Extend MLNX_DriverCapabilities 49
 - 5.10.3 MLNX_DriverEthCapabilities – Extend MLNX_DriverCapabilities 50
 - 5.10.4 MLNX_DriverIbCapabilities – Extend MLNX_DriverCapabilities 51

5.10.5	MLNX_DriverIpOverIbCapabilities – Extend MLNX_DriverCapabilities	51
5.10.6	MLNX_DriverCoreSettingData – Extend MLNX_DriverSettingData.....	52
5.10.7	MLNX_DriverEthSettingData – Extend MLNX_DriverSettingData	54
5.10.8	MLNX_DriverIbSettingData – Extend MLNX_DriverSettingData	54
5.10.9	MLNX_DriverIpOverIbSettingData – Extend MLNX_DriverSettingData	55
5.10.10	MLNX_DriverElementCapabilities – Extend CIM_ElementCapabilities	56
5.10.11	MLNX_DriverElementSettingData – Extend CIM_ElementSettingData	56

List of Figures

Figure 1: Device Level CIM Classes	14
Figure 2: Network Adapter Level View	15
Figure 3: Software Identity View	16
Figure 4: Software Update View	17
Figure 5: CDM View	18
Figure 6: Driver Level Management View	19
Figure 7: Indications View	20

List of Tables

Table 1: CIM cmdlets Description.....	21
Table 2: MLNX_RegisteredProfile Class Properties	25
Table 3: MLNX_ElementConformsToProfile Associations	25
Table 4: MLNX_SoftwareIdentity Class Properties	26
Table 5: MLNX_FirmwareIdentity Class Properties	27
Table 6: MLNX_DriverIdentity Class Properties.....	28
Table 7: MLNX_ElementSoftwareIdentity Associations	28
Table 8: MLNX_InstalledSoftwareIdentity Associations.....	29
Table 9: MLNX_SoftwareInstallationService Properties	29
Table 10: InstallFromSoftwareIdentity Method Parameters	30
Table 11: InstallFromURI Method Parameters.....	30
Table 12: Install Method Parameters	30
Table 13: Extend CIM_SoftwareInstallationService Properties.....	31
Table 14: InstallFromSoftwareIdentity Method Parameters	31
Table 15: InstallFromURI Method Parameters.....	32
Table 16: Update Method Parameters	32
Table 17: MLNX_PCIDevice Class Properties	32
Table 18: RequestStateChange Method Parameters	34
Table 19: MLNX_SystemDevice Associations	34
Table 20: MLNX_PCIDeviceCapabilities Class Properties	34
Table 21: MLNX_PCIDeviceSettingData Class Properties	35
Table 22: MLNX_PCIDeviceSriovSettingData Class Properties.....	35
Table 23: Set Value Method Parameters	36
Table 24: MLNX_PCIDevicePortTypeSettingData Class Properties	36
Table 25: Set Value Method Parameters	36
Table 26: MLNX_IBPort Class Properties	37
Table 27: MLNX_IBPortCounters Class Properties	38
Table 28: MLNX_IBPortControlledBy Associations.....	39
Table 29: MLNX_NetAdapter Class Properties.....	39
Table 30: MLNX_NetAdapter Class Methods	40
Table 31: MLNX_NetAdapterGenStatistics Class Properties	40
Table 32: MLNX_NetAdapterRoceSettingData Class Properties	42
Table 33: MLNX_NetAdapterControlledBy Associations	42
Table 34: MLNX_NetAdapterFlowControlSettingData Class Properties.....	42
Table 35: MLNX_CardClass Properties	43
Table 36: MLNX_Realizes Associations	44
Table 37: MLNX_DiagnosticTestHca Class Properties.....	45
Table 38: RunDiagnostic Method Parameters	46
Table 39: RunDiagnosticService Method Parameters	46
Table 40: MLNX_DiagnosticConcreteJob Class Properties.....	46
Table 41: Extend CIM_DiagnosticJob Class Properties	47
Table 42: MLNX_AvailableDiagnosticService Associations.....	47

Table 43: MLNX_DiagnosticUseOfLog Associations 47

Table 44: MLNX_DriverService Class Properties 48

Table 45: MLNX_DriverCoreCapabilities Class Properties 49

Table 46: MLNX_DriverEthCapabilities Class Properties 50

Table 47: MLNX_DriverIbCapabilities Class Properties 51

Table 48: MLNX_DriverIpOverIbCapabilities Class Properties 51

Table 49: MLNX_DriverCoreSettingData Class Properties 52

Table 50: SetValue Method Parameters 53

Table 51: MLNX_DriverEthSettingData Class Properties 54

Table 52: SetValue Method Parameters 54

Table 53: MLNX_DriverIbSettingData Class Properties 54

Table 54: SetValue Method Parameters 55

Table 55: MLNX_DriverIpOverIbSettingData Class Properties 55

Table 56: SetValue Method Parameters 55

Table 57: MLNX_DriverElementCapabilities Associations 56

Table 58: MLNX_DriverElementSettingData Associations 56

About this Manual

The purpose of this document is to provide information for Common Information Model (CIM) Provider for Mellanox ConnectX® adapter cards family. CIM providers use a standard way to receive adapter software identity information, apply firmware and software upgrade, get diagnostic information, and more.

1 Overview

Distributed enterprise computing has drastically changed the way users work. It allows people to access various types of data anytime and anywhere. This can be achieved by using a technology which can provide optimal management for most of the entities in the enterprise network and can co-exist with the existing technologies.

The Common Information Model (CIM) is an open standard that defines how managed elements in an IT environment are represented as a common set of objects and relationships between them. This is intended to allow consistent management of these managed elements, independent of their manufacturer or provider. CIM provides a unified method for managing an enterprise network without requiring an overhaul of the existing network management infrastructure.

CIM not only represents managed elements and management information, but also provides means to actively control and manage these elements using any standard CIM-based management software. The CIM standard is defined and published by the Distributed Management Task Force (DMTF). A related standard is Web-Based Enterprise Management (WBEM, also defined by DMTF) which defines a particular implementation of CIM, including protocols for discovering and accessing such CIM implementations.

CIM Providers allow adapters management using any standard CIM-based management software.

WMI is Microsoft's implementation of CIM CIMOM (CIM Object Manager). While WMI protocol is proprietary "DCOM", Windows 8 protocol is based on the standard OpenWebm. Moreover, Windows 8 introduces PowerShell CIM cmdlets, acts as CIM client and enables interoperability over different Oss.

1.1 CIM Management Flow

CIM Management flow occurs in the following way:

1. CIM Server receives CIM operation requests from a CIM client.
2. CIM Server decodes the requests, coordinates the processing of requests and the responses among the providers, and sends the CIM operations back to the CIM client.
3. CIM Provider processes the CIM operations on managed resources.
4. CIM Provider translates CIM-formatted requests into resource-specific operations and vice versa.
5. CIM Provider provides the mapping between the CIM interface and the resource-specific interface and contains the implementation for a set of CIM operations for a defined set of managed resources.

1.2 General Design and Implementation Guidelines

- CIM Provider implementation should be as much independent as possible of Mellanox Tools and Driver capabilities
- CIM Providers should use underlying OS tools

- CIM Provider should encapsulate common, OS independent logic into common module. This should include workflow framework with defined entry points for OS specific calls.
- CIM Providers should implement OS specific logic/methods in dedicated module that will be called by common module
- CIM Provider should provide means for logging and debugging purposes

2 CIM Support Requirements and Settings

2.1 Supported CIM Profiles

- DSP1033 - [Profile Registration](#) (on page 25)
- DSP1023 - [Software Inventory Profile](#) (on page 26)
- DSP1025 - [Software Update Profile](#) (on page 29)
- DSP1075 - [PCI Device Profile](#) (on page 32)
- DSP1014 - [Ethernet Port Profile](#) (on page 39)
- DSP1011 - [Physical Asset Profile](#) (on page 42)
- DSP1035 - [Host LAN Network Port Profile](#) (on page 44)
- DSP1002 - [Diagnostics CDMv2 Profile](#) (on page 44)
- DSP1054 - [Indications Profile](#) (on page 48)
- [Drivers Services Profile](#) (on page 48)

2.2 Supported CIM Operations

The list of Mellanox CIM provider supported operations is as following:

- GetInstance
- Associators
- AssociatorNames
- References
- ReferenceNames
- EnumerateInstances
- EnumerateInstanceNames
- InvokeMethod

2.3 CIM Object Manager (CIMOM) Requirements

Mellanox CIM provider should support:

- Windows WMI

2.4 Supported Operating Systems

The following OSs are supported:

- Windows 2008 R2 with Windows Management Framework 3.0
- Windows Server 2012
- Windows Server 2012 R2

2.5 Configuring CIM Provider

The CIM Provider is installed by default as part of WinOF package installation. For further details, please refer to WinOF User Manual.

2.5.1 Remote Execution

➤ *To enable remote execution, while the default WMI protocol is WSMAN:*

```
> winrm quickconfig
```

➤ *To verify the execution policy (if needed):*

```
> Set-ExecutionPolicy RemoteSigned
```

➤ *To change the protocol to legacy DCOM:*

- Use CimSessionOption cmdlets

2.5.2 Software Installation

Wbem runs "NETWORK SERVICE", therefore, you may need to configure your administration group or set the Windows Installer component to allow Software Installation using CIM.



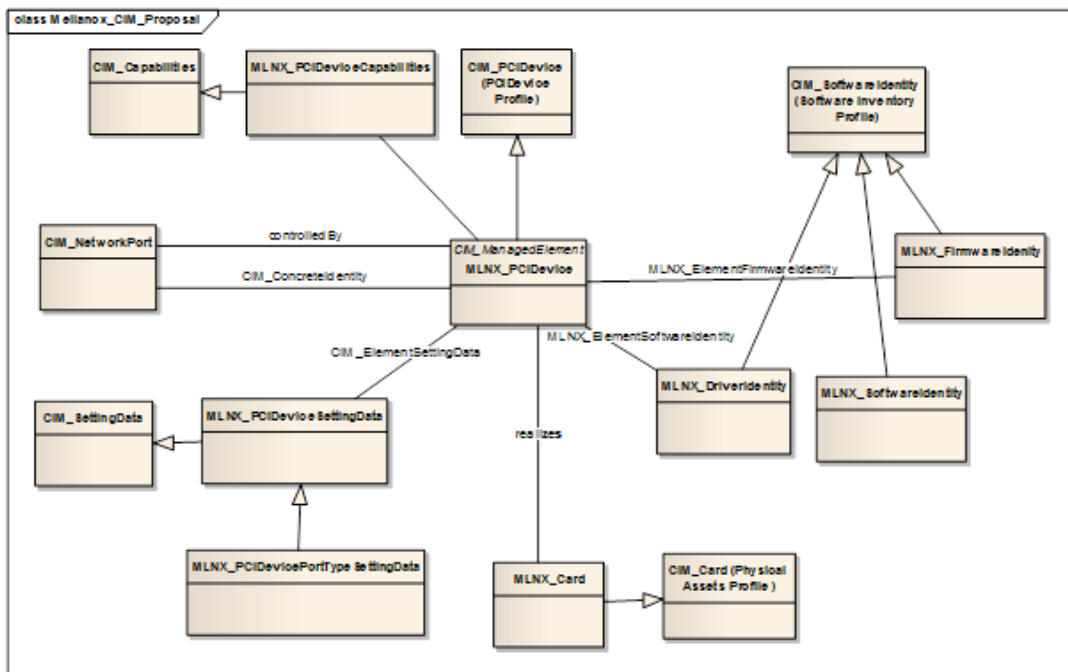
NOTE: All modification methods require Administration rights.

3 Data Model

3.1 Device Level View

The Device Level view represents the CIM classes that are in use in order to represent a network port associated controller, its settings and capabilities. Associations with the Controller’s physical aspects, software image and profile implementation version information are also modeled here. The device level view displays the Mellanox extended CIM classes that provide management aspects in conformance with PCI Device profile.

Figure 1: Device Level CIM Classes



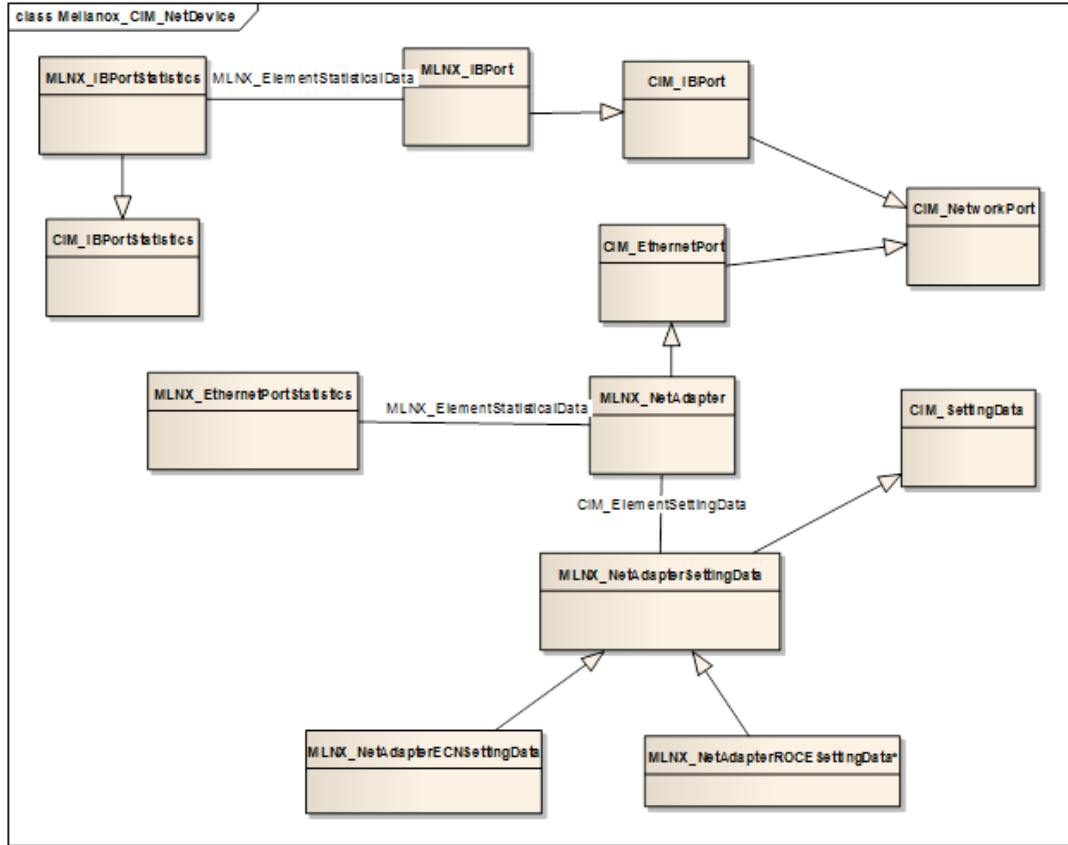
3.2 Network Adapter (Port) Level View

The Network Adapter Level view represents the CIM classes that are in use in order to represent a network port that provides a LAN interface in a managed system. The view includes a specification of the network port, both for Ethernet and InfiniBand ports. Different attributes settings are also modeled here. There are dedicated CIM classes that provide statistics information for both Ethernet and InfiniBand ports.

In the future the associated network endpoint and the realization of the connection in a physical connector will be supported.

The Network Adapter level view displays the Mellanox extended CIM classes that provide management aspects in conformance with The Host LAN Network Port Profile and Ethernet Port Profile.

Figure 2: Network Adapter Level View

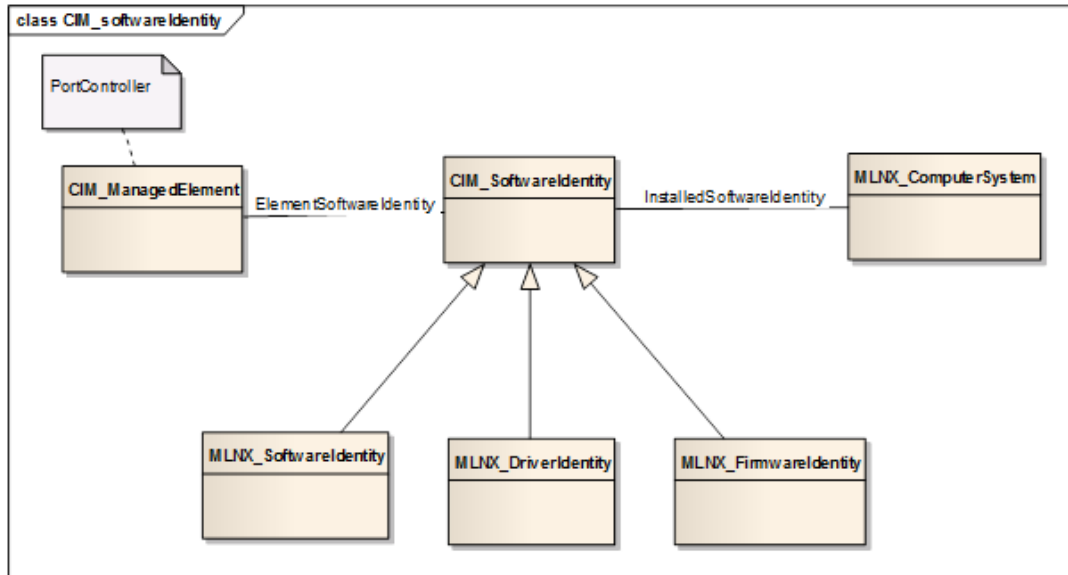


*MLNX_NetAdapterROCESettingData class applies to WinOF v4.40 only.

3.3 Software Inventory View

The Software Inventory view represents the CIM classes that are in use in order to represent an inventory of installed firmware and Drivers and related software in a managed system. The Software Inventory view displays the Mellanox extended CIM classes that provide management aspects in conformance with Software Inventory Profile.

Figure 3: Software Identity View

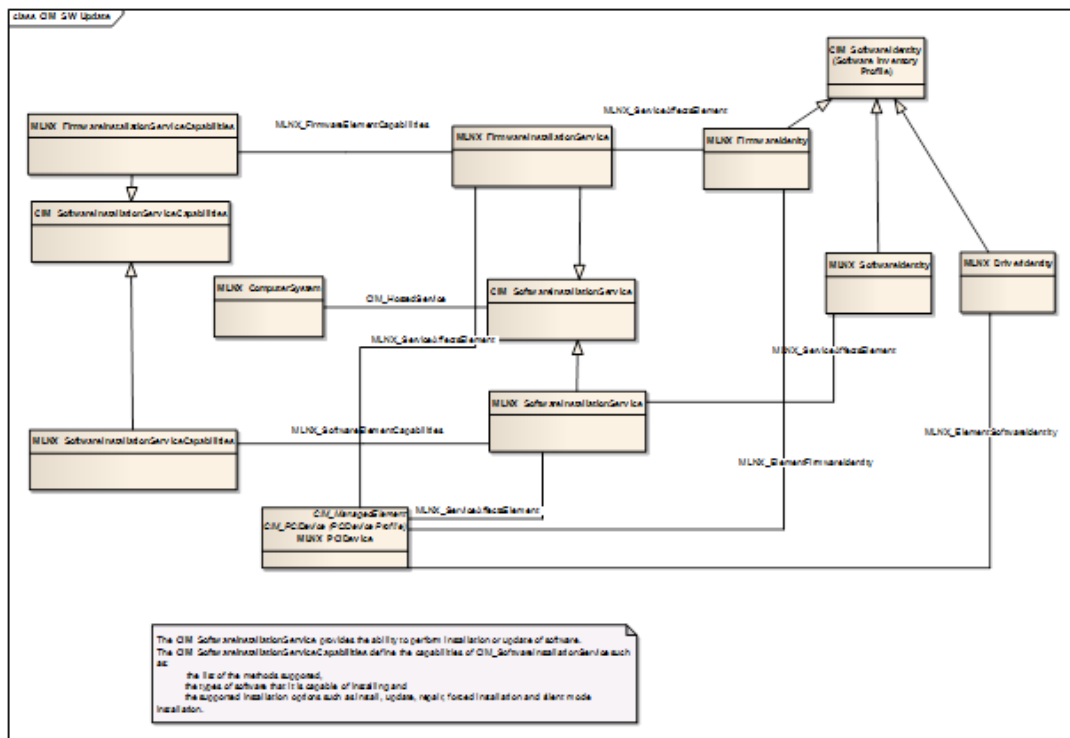


3.4 Software Update View

The Software Update view represents the CIM classes that are in use in order to support the installation and update of firmware and drivers software on a HCA. There are two Software Installation Service classes, one for managing firmware update and other for managing drivers' software update processes. Both service capabilities are modeled in separate classes correlating to the Service.

The Software Update view displays the Mellanox extended CIM classes that provide management aspects in conformance with Software Update Profile.

Figure 4: Software Update View

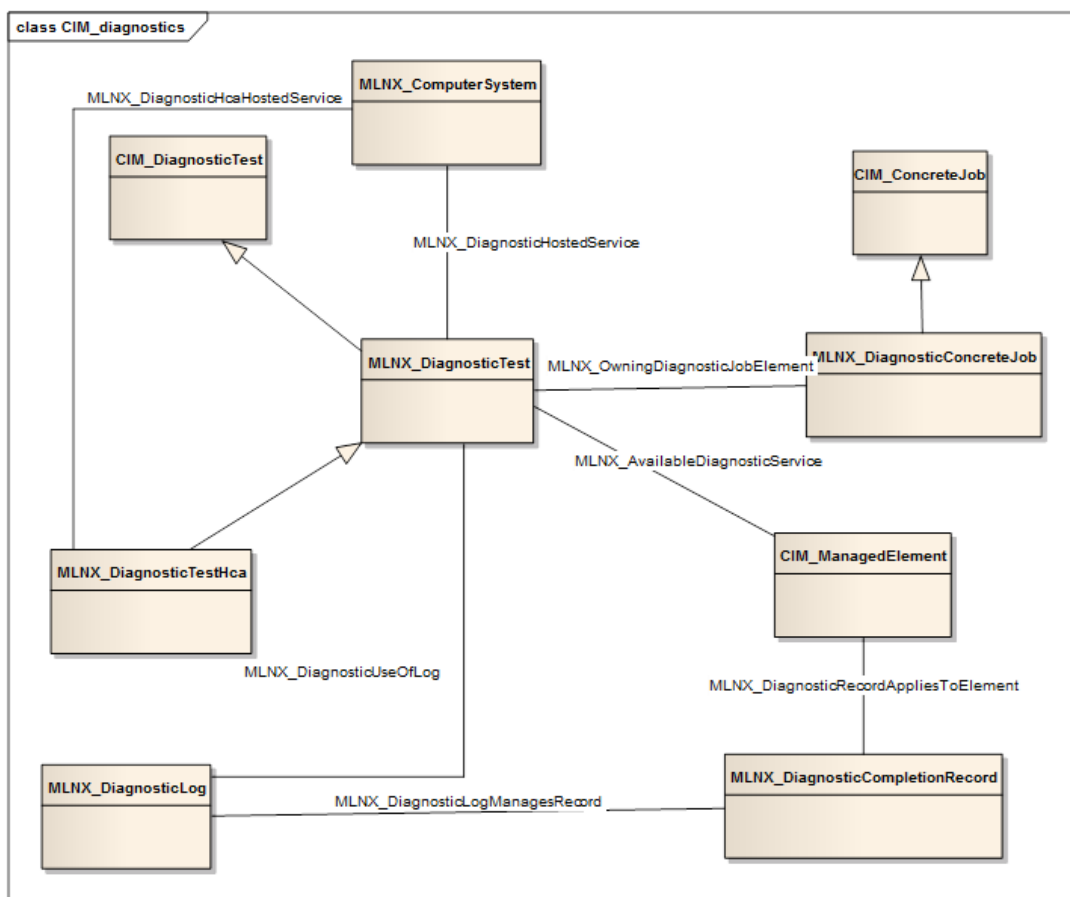


3.5 Common Diagnostic Model (CDM) View

The Common Diagnostic Model (CDM) view represents the CIM classes that are in use in order to support the Common Diagnostic Model v2 required management. The CDM View displays classes that provide the capability to run diagnostic services in a managed system. It includes a specification of the Diagnostic Test Service, its configuration, its associated capabilities and its logging mechanisms. The instance of the MLNX_DiagnosticTest is a central instance of this profile.

The CDM view displays the Mellanox extended CIM classes that provide management aspects in conformance with Diagnostics Profile.

Figure 5: CDM View



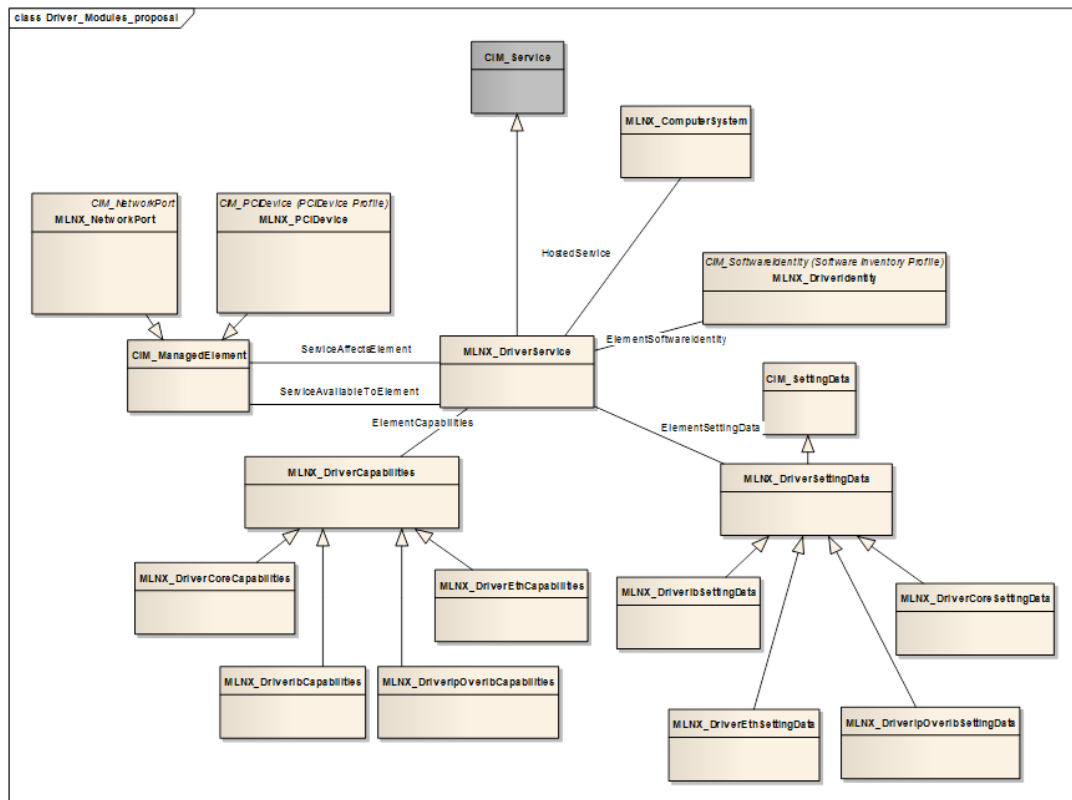
3.6 Driver Level Management View

The below model represents CIM classes, that are used to manage OS driver capabilities and settings.

- Following this general Model, for each driver layer there is the deriving classes from:
 - MLNX_DriverService – stands for the managed element responsible to manage Drivers settings and retrieve its capabilities.
 - MLNX_DriverCapabilities – stands for Driver Layer managed element that holds Driver Layer capabilities.

- MLNX_DriverSettingData – stands for Driver Layer managed element that holds requested driver layer configurations and operational parameters for.
- Each Driver Layer Software image is presented by MLNX_DriverIdentity (with version details).
- The following are different driver layers that are managed in Windows environment:
 - mlx4_bus
 - ibbus
 - mlx4eth63

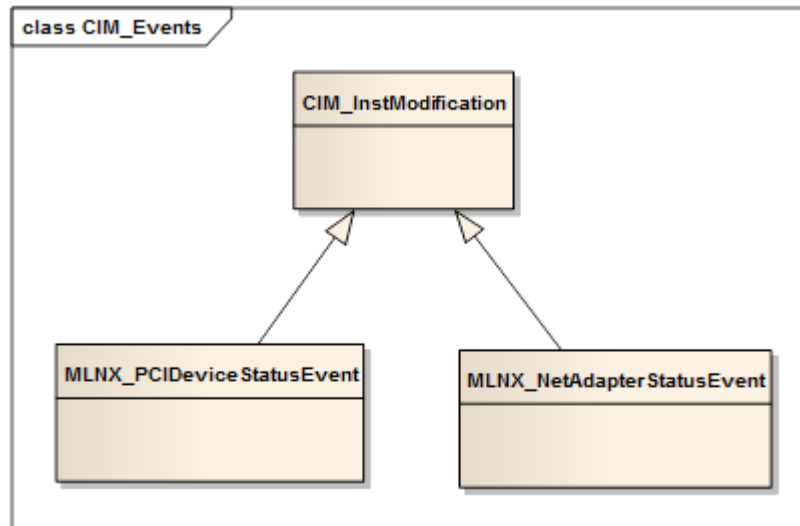
Figure 6: Driver Level Management View



3.7 CIM Indications View

The Indications view represents the CIM classes that are in use in order to provide support for Indications that can be advertised by Mellanox CIM provider. Their detailed attributes of each event are defined in next sections of this document.

Figure 7: Indications View



4 Using CIM Features

4.1 Overview

The current CIM provider was build using Microsoft MI API. It support WMI OMI and conform to the Common Manageability Interface (CMPI).

In this chapter we will use the new PowerShell CIM cmdlets to demonstrate the usability of Mellanox CIM provider. Other CIMOMs such as OMI, WMI or OpenPegasus (remotely) are not covered in this document.

4.2 Using CIM cmdlets

Table 1: CIM cmdlets Description

Cmdlet	Description
Get-CimAssociatedInstance	Gets the CIM instances that are connected to a specific CIM instance by an association.
Get-CimClass	Retrieves a list of CIM classes in a specific namespace.
Get-CimInstance	Gets the CIM instances of a class from a CIM server.
Get-CimSession	Gets the CIM session objects from the current session.
Invoke-CimMethod	Invokes a method of a CIM class.
New-CimInstance	Creates a CIM instance.
New-CimSession	Creates a CIM session.
New-CimSessionOption	Specifies advanced options for the New-CimSession cmdlet.
Register-CimIndicationEvent	Subscribes to indications using a filter expression or a query expression.
Remove-CimInstance	Removes a CIM instance from a computer.
Remove-CimSession	Removes one or more CIM sessions.
Set-CimInstance	Modifies a CIM instance on a computer.

For more details on using CIM cmdlets, please follow the link below:

<http://technet.microsoft.com/en-us/library/jj553783.aspx>

[Configure Remote Management in Server Manager](#)

CIM cmdlets communicate via both the WSMAN and DCOM protocols and WMI communicates only via DCOM.

WSMan protocol is also much faster and also more secure in comparison to DCOM. Also by using WSMAN as a common protocol to communicate with remote devices we can bypass most of firewall issue which we may encounter which are not the case with DCOM as they are prone to firewall issues.



NOTE: You can run the PowerShell CIM Cmdlets from Windows Server® 2008 R2 or Windows® 7 client systems by installing Windows Management Framework 3.0, available for download here <http://www.microsoft.com/en-us/download/details.aspx?id=34595>. Thus, the features - Interoperability and backward compatibility, make the usage of PowerShell CIM Cmdlets really flexible and powerful.

4.3 Namespace

- The Mellanox Provider extend the standard CIM schemas and use the following namespace: Root\standardcimv2\mlnx
- All classes start with the MLNX_ prefix.
- All cmdlets start with Mlnx prefix.

Example:

```
MLNX_IBPort extends CIM_IBPort and correlates to MlnxIBPort cmdlet.
```

➤ *To view all CIM classes in Mellanox namespace:*

```
PS C:\> Get-CimClass -Namespace root/StandardCimv2/mlnx
```

4.4 Establishing the Session

1. Set up a session to the remote system.

This session is not persistent across OS reboots.

To set up the session, you need to set the CIM Session options with the Cmdlet New-CimSessionOption.

Example Using WSMAN:

```
PS C:\Users\Administrator> $cimop=New-CimSessionOption -SkipCACheck
-SkipCNCheck -SkipRevocationCheck -Encoding Utf8 -UseSsl
```

Example Using DCOM:

```
PS C:\Users\Administrator> $cimop=New-CimSessionOption -Protocol Dcom
```

Once the session options are set and saved to a variable, you can go ahead and create a session to the remote system. This can be accomplished with the New-CIMSession Cmdlet.

The New-CimSession cmdlet creates a CIM session. A CIM session is a client-side object representing a connection to a local computer or a remote computer. The CIM session contains information about the connection, such as ComputerName, the protocol used for the connection, session ID and instance ID.

Example:

```
PS C:\ > $IPAddress="xxx"
PS C:\ > $Username="USER"
PS C:\ > $session=New-CimSession -Authentication Basic -Credential $Username
-ComputerName $IPAddress -Port 443 -SessionOption $cimop
PS C:\ > echo $session
```

Output:

```
Id           : 1
Name        : CimSession1
InstanceId  : e8615b82-bb22-4ec4-9377-a86c03ba4995
ComputerName : 10.*.*.*
Protocol    : WSMAN
```

Example (multiple servers):

```
$CimSession = New-CimSession -ComputerName 'server1', 'server2', 'server3'
-SessionOption $cimop
```

The remote IP Address and the credentials should be provided for the system, and thus setting up a session with Basic authentication. Once the session is established, you are ready to remotely manage your server.

4.5 Get Instance

The Get-CimInstance cmdlet gets the CIM instances of a class from a CIM server. You can specify either the classname or a query for this cmdlet.

Example:

```
PS C:\ > Get-CimInstance -CimSession $Session -Namespace
root/standardcimv2/mlnx -Classname MLNX_NetAdapter -property name,caption
```

Example:

```
PS C:\ > Get-CimInstance -CimSession $Session -Namespace
root/standardcimv2/mlnx Query "SELECT * from MLNX_NetAdapter WHERE name LIKE '
Ethernet M%'"
```

4.6 Invoke Method

The Invoke-CimMethod cmdlet invokes a method of a CIM class or CIM instance using the name-value pairs specified by the Arguments parameter.

Example (Enable RoCE option in WinOF 4.40):

```
PS C:\> $inst = Get-CimInstance -CimSession $session -Namespace
root/standardcimv2/mlnx -Query "SELECT * from MLNX_NetAdapterRoceSettingData
WHERE Enabled = 'TRUE'"
PS C:\ > Invoke-CimMethod -InputObject $inst -CimSession $session -MethodName
"Disable"
```

Example (Set service RoCE mode in WinOF 4.55 and above):

```
PS C:\ > Invoke-CimMethod -InputObject $inst -CimSession $session -MethodName
"SetValue" -Arguments @{RoceMode ="1.0"}
```

Example (Set service value):

```
PS C:\ > $inst = Get-CimInstance -CimSession $session -Namespace
root/standardcimv2/mlnx -Query "SELECT * from MLNX_DriverCoreSettingData WHERE
LogNumCq = '16'"
PS C:\ > Invoke-CimMethod -InputObject $inst -CimSession $session -MethodName
"SetValue" -Arguments @{LogNumCq ="20"}
```

Example (Restore to service default):

```
PS C:\ > Invoke-CimMethod -InputObject $inst -CimSession $session -MethodName
"SetDefault"
```

- *To view the method parameters, types, and qualifiers:*

```
PS C:\ > $class = Get-CimClass -CimSession $session -Namespace
root/standardcimv2/mlnx -ClassName MLNX_FirmwareInstallationService
PS C:\ > $class.CimClassMethods['Update'].Parameters
```

4.7 Association

- *To use PowerShell custom cmdlets:*

- Pipe the cmdlets of the same family.

Example:

```
PS C:\> Get-MlnxAdapter -Name "Ethernet 6" | Enable-MlnxNetAdapterRoceSetting
```

- *To retrieve the CIM instances connected to a specific CIM instance:*

- Use the source instance `Get-CimAssociatedInstance`.

Example:

```
PS C:\ > $instance = Get-CimInstance -CimSession $session -Namespace
root/standardcimv2/mlnx -ClassName MLNX_PCIDevice -Filter "Name='HCA 0'"
PS C:\ > Get-CimAssociatedInstance -InputObject $instance -Association
CIM_ConcreteIdentity
```

4.8 Events' Registry

- *To subscribe to an indication using an indication class or a query expression, use the `Register-CimIndicationEvent` command:*

1. Use the `SourceIdentifier` parameter.
2. Name the subscription.

Example (register the event):

```
PS C:\ > Register-CimIndicationEvent -class MLNX_PCIDeviceStatusEvent
-namespace root/StandardCimv2/mlnx -SourceIdentifier "PciStatus"
PS C:\ > Disable-MlnxPCIDevice -Name "HCA 0"
PS C:\ > $event = Get-Event -SourceIdentifier PciStatus
PS C:\ > $event.SourceEventArgs.NewEvent.PreviousInstance | Select Name,
Status
PS C:\ > $event.SourceEventArgs.NewEvent.SourceInstance | Select Name, Status
```

Example (unregister the event):

```
PS C:\ > Unregister-Event *
```

4.9 Custom Cmdlets

Some of the classes and method are displayed as PowerShell custom Cmdlets.

All custom cmdlets support the `CimSession` parameter.

- *To view the list of the available methods:*

```
PS C:\ > Get-Command -Module MLNXProvider
```

- *To view the method syntax:*

```
PS C:\ > Get-Command <method> -syntax
```


5 Profile Classes, Methods and Properties

5.1 Profile Registration

The Profile Registration describes the necessary properties and methods to represent profile and profile versioning implementation conformance. Profile Registration defines the classes used to describe the DMTF profile registration and the version information of the profiles advertised as implemented for a managed system and components of the system.

This profile specifies how to identify unambiguously the classes, properties, methods, and values that must be instantiated to represent the profile name, version, and owning organization information that is modeled using the DMTF CIM Schema.

The Profile Registration describes the registration and versioning of Common Information Model (CIM) profiles that are implemented by CIM-based system and component-management instrumentation.

5.1.1 MLNX_RegisteredProfile – Extend CIM_RegisteredProfile

5.1.1.1 MLNX_RegisteredProfile Class Properties

Table 2: MLNX_RegisteredProfile Class Properties

Name	Data Type	Expected Sample
Caption	uint16	Mellanox PCI Attached Devices
Description	uint16	Profile of all Mellanox Technologies' PCI attached devices.
InstanceID	string	MLNX:DSP1075:PCIDeviceProfile
RegisteredName	string	PCI Device
RegisteredOrganization	uint64	2
RegisteredVersion	uint16	1.0.0

5.1.2 MLNX_ElementConformsToProfile – Extend CIM_ElementConformsToProfile

5.1.2.1 MLNX_ElementConformsToProfile Associations

Table 3: MLNX_ElementConformsToProfile Associations

Name	Reference Class
ConformantStandard	MLNX_RegisteredProfile
ManagedElement	CIM_ManagedElement

5.2 Software Inventory Profile

The Software Inventory Profile describes the CIM schema elements required to provide an inventory of installed BIOS, firmware, drivers, and related software in a managed system.

This profile also describes the CIM schema elements required to represent the software that can be installed on a managed system.

The following Network Adapter SW that presented by this profile is:

- Firmware SW
- Driver SW
- Diagnostic SW
- CIM Provider SW

5.2.1 MLNX_ SoftwareIdentity – Extend CIM_ SoftwareIdentity

5.2.1.1 MLNX_ SoftwareIdentity Class Properties

Table 4: MLNX_ SoftwareIdentity Class Properties

Name	Data Type	Expected Sample
BuildNumber	uint16	14370
Caption	string	MLNX_SoftwareIdentity 'Mellanox Software stack for RDMA aware devices'
InstanceID	string	MLNX_VPI
IsEntity	boolean	False
IsLargeBuildNumber	boolean	False
Description	string	Mellanox Software stack for RDMA aware devices
MajorVersion	uint16	4
Manufacturer	string	Mellanox Technologies
ElementName	string	MLNX_VPI
Name	string	MLNX_VPI
Status	string	OK
OperationalStatus	uint16[]	{2}
MinorVersion	uint16	41
StatusDescriptions	string[]	{OK}
InstallLocation	string	C:\Program Files\Mellanox\MLNX_VPI\
RevisionNumber	uint16	0
VersionString	string	4.41.14370
ClassificationDescriptions	string[]	{Software Bundle}
Classifications	uint16[]	{13}
IdentityInfoType	string[]	{SoftwareID, SoftwareStatus}
IdentityInfoValue	string[]	{MLNX_VPI, 2}

Name	Data Type	Expected Sample
Languages	string[]	{English (United States)}
TargetOperatingSystems	string[]	
TargetOSTypes	uint16[]	
TargetTypes	string[]	

5.2.2 MLNX_ FirmwareIdentity – Extend CIM_SoftwareIdentity

5.2.2.1 MLNX_ FirmwareIdentity Class Properties

Table 5: MLNX_ FirmwareIdentity Class Properties

Name	Data Type	Expected Sample
BuildNumber	uint16	
Caption	string	MLNX_FirmwareIdentity 'Firmware for device 26418 with PSID MT_0D20110008'
InstanceID	string	mt26418_pci_cr0
IsEntity	boolean	False
IsLargeBuildNumber	boolean	False
Description	string	Firmware for device 26418 with PSID MT_0D20110008
MajorVersion	uint16	2
Manufacturer	string	Mellanox Technologies
ElementName	string	MT_0D20110008
Name	string	MLNX VPI Adapter MT1008X01108 firmware
Status	string	OK
OperationalStatus	uint16[]	{2}
MinorVersion	uint16	11
StatusDescriptions	string[]	{OK}
Location	string	2:0:0
RevisionNumber	uint16	820
SerialNumber	string	MT1008X01108
VersionString	string	2.11.820
ClassificationDescriptions	string[]	{Firmware}
Classifications	uint16[]	{10}
IdentityInfoType	string[]	{FirmwareID, FirmwareStatus}
IdentityInfoValue	string[]	{mt26418_pci_cr0, 2}
ExtendedResourceType	uint16	1

5.2.3 MLNX_DriverIdentity – Extend CIM_SoftwareIdentity

5.2.3.1 MLNX_DriverIdentity Class Properties

Table 6: MLNX_DriverIdentity Class Properties

Name	Data Type	Expected Sample
BuildNumber	uint16	14370
Caption	string	MLNX_DriverIdentity 'Mellanox ConnectX based IPoIB Adapter (NDIS 6.3) Driver'
InstanceID	string	ipoib6x
IsEntity	boolean	False
IsLargeBuildNumber	boolean	False
Description	string	Mellanox ConnectX based IPoIB Adapter (NDIS 6.3) Driver
MajorVersion	uint16	4
Manufacturer	string	Mellanox Technologies
ElementName	string	ipoib6x
Name	string	ipoib6x
Status	string	OK
OperationalStatus	uint16[]	{2}
MinorVersion	uint16	41
StatusDescriptions	string[]	{OK}
RevisionNumber	uint16	0
VersionString	string	4.41.14370.0
ClassificationDescriptions	string[]	{Driver}
Classifications	uint16[]	{2}
IdentityInfoType	string[]	{SoftwareID, SoftwareStatus}
IdentityInfoValue	string[]	{ipoib6x, 2}
Languages	string[]	{English (United States)}
InstallLocation	string	c:\Windows\system32\DRIVERS\ipoib6x.sys

5.2.4 MLNX_ElementSoftwareIdentity – Extend CIM_ElementSoftwareIdentity

5.2.4.1 MLNX_ElementSoftwareIdentity Associations

Table 7: MLNX_ElementSoftwareIdentity Associations

Name	Reference Class
Antecedent	CIM_SoftwareIdentity
Dependent	CIM_ManagedElement

5.2.5 MLNX_InstalledSoftwareIdentity – Extend CIM_InstalledSoftwareIdentity

5.2.5.1 MLNX_InstalledSoftwareIdentity Associations

Table 8: MLNX_InstalledSoftwareIdentity Associations

Name	Reference Class
InstalledSoftware	CIM_SoftwareIdentity
System	CIM_System

5.3 Software Update Profile

The Software Update Profile describes the classes, associations, properties, and methods used to support the installation and update of BIOS, firmware, drivers and related software on a managed element within a managed system.

The profile defines relationship between a managed element and the installation service that represents the availability of software installation and update functionality for a managed element.

CIM_SoftwareInstallationService is a central class of this profile.

Following use case should be supported:

- Find Software Installation Services that Can Install or Update Software
- Find Software Available for Installation
- Find the Software Installation Services compatible with a Software Identity
- Determine Whether Installing a Software Identity Requires a Reboot
- Install or Update Software on a Managed Element Using URI or Software Identity

Mellanox CIM Provider supports Software and firmware software update

The installation of the software bundles (WinOf and WinMFT) requires Windows Installer Service enabling for “Network Service” users. While the InstallFromURI is conformed to the profile definition, the InstallFromSoftwareIdentity relay on a configuration file in the provider root path.

The method reads the SoftwareIdentity.txt file and searches for WinOf and WinMFT image ready for installation which the method expects to receive as a reference object.

5.3.1 MLNX_SoftwareInstallationService – Extend CIM_SoftwareInstallationService

5.3.1.1 MLNX_SoftwareInstallationService Class Properties

Table 9: MLNX_SoftwareInstallationService Properties

Name	Data Type	Expected Sample
Caption	string	Mellanox Firmware Tools for Windows
CreationClassName	string	MLNX_SoftwareInstallationService
Description	string	Mellanox Firmware Tools

Name	Data Type	Expected Sample
ElementName	string	Mellanox WinMFT
EnabledDefault	uint16	2
EnabledState	uint16	5
InstanceID	string	MLNX:WinMFT
Name	string	WinMFT64
PrimaryOwnerContact	string	http://www.mellanox.com
PrimaryOwnerName	string	Mellanox Technologies
RequestedState	uint16	12
Status	string	OK
SystemCreationClassName	string	CIM_System
SystemName	string	WINDOWS-124SFT7

5.3.1.2 MLNX_SoftwareInstallationService Class Methods

5.3.1.2.1 InstallFromSoftwareIdentity (uint32)

Table 10: InstallFromSoftwareIdentity Method Parameters

Name	Qualifier	Type
InstallOptions	IN	uint16[]
InstallOptionsValues	IN	string[]
Collection	IN	REF CIM_Collection
Job	OUT	REF CIM_ConcreteJob
Source	IN	REF CIM_SoftwareIdentity
Target	IN	REF CIM_ManagedElement

5.3.1.2.2 InstallFromURI (uint32)

Table 11: InstallFromURI Method Parameters

Name	Qualifier	Type
URI	IN	string
InstallOptions	IN	uint16[]
InstallOptionsValues	IN	string[]
Job	OUT	REF CIM_ConcreteJob
Target	IN	REF CIM_ManagedElement

5.3.1.2.3 Install (uint32)

Table 12: Install Method Parameters

Name	Qualifier	Type
Source	IN	string
Target	IN	string

Name	Qualifier	Type
Install	IN	boolean
Update	IN	boolean
Repair	IN	boolean
Uninstall	IN	boolean
Log	IN	string
Reboot	IN	boolean

5.3.2 MLNX_FirmwareInstallationService – Extend CIM_SoftwareInstallationService

5.3.2.1 Class Properties

Table 13: Extend CIM_SoftwareInstallationService Properties

Name	Data Type	Expected Sample
Caption	string	MLNX_FirmwareInstallationService 'Mellanox Firmware Installation Service'
CreationClassName	string	MLNX_FirmwareInstallationService
Description	string	Mellanox Firmware Installation Service
ElementName	string	Mellanox Firmware Installation Service
EnabledDefault	uint16	2
EnabledState	uint16	5
InstanceID	string	MLNX:Firmware Installation Service
Name	string	MLNX:Firmware Installation Service
PrimaryOwnerContact	string	http://www.mellanox.com
PrimaryOwnerName	string	Mellanox Technologies
RequestedState	uint16	12
SystemCreationClassName	string	CIM_System
SystemName	string	WINDOWS-124SFT7

5.3.2.2 Class Methods

5.3.2.2.1 InstallFromSoftwareIdentity (uint32)

Table 14: InstallFromSoftwareIdentity Method Parameters

Name	Qualifier	Type
InstallOptions	IN	uint16[]
InstallOptionsValues	IN	string[]
Collection	IN	REF CIM_Collection
Job	OUT	REF CIM_ConcreteJob
Source	IN	REF CIM_SoftwareIdentity

Name	Qualifier	Type
Target	IN	REF CIM_ManagedElement

5.3.2.2.2 InstallFromURI (uint32)

Table 15: InstallFromURI Method Parameters

Name	Qualifier	Type
URI	IN	string
InstallOptions	IN	uint16[]
InstallOptionsValues	IN	string[]
Job	OUT	REF CIM_ConcreteJob
Target	IN	REF CIM_ManagedElement

5.3.2.2.3 Update (uint32)

Table 16: Update Method Parameters

Name	Qualifier	Type
Source	IN	string
Device	IN	string
Reboot	IN	boolean
Force	IN	boolean
Log	IN	string

5.4 PCI Device Profile

Logical PCI devices in the computer system are represented using CIM_PCIDevice. The PCI Device Profile adding the capability to represent PCI devices for manageability, including PCI, PCI-X, PCI Express, bridge and switch devices. The PCI device as a logical device is modeled as referencing the physical package for physical asset information and profile versioning for the schema implementation version information. CIM_PCIDevice implement the properties according the Profile definition. CIM_ConcreteIdentity is used to associate an instance of CIM_LogicalDevice (EthernetPort/IBPort) with an instance of CIM_PCIDevice of which the CIM_LogicalDevice instance represents an alternate aspect of the PCI device.

5.4.1 MLNX_PCIDevice – Extend CIM_PCIDevice

5.4.1.1 Class Properties

Table 17: MLNX_PCIDevice Class Properties

Name	Data Type	Expected Sample
BusNumber	uint8	1
DeviceNumber	uint8	0
FunctionNumber	uint8	0
MaxLatency	uint8	0

Name	Data Type	Expected Sample
MinGrantTime	uint8	0
PCIDeviceID	uint16	4099
RevisionID	uint8	0
SubsystemID	uint16	5555
SubsystemVendorID	uint16	
VendorID	uint16	5555
Availability	uint16	3
Caption	string	MLNX_PCIDevice 'Mellanox ConnectX-3 (MT04099) Network Adapter'
CreationClassName	string	MLNX_PCIDevice
Description	string	Mellanox ConnectX-3 (MT04099) Network Adapter
DeviceID	string	PCI\VEN_15B3&DEV_1003&SUBSYS_005015B3&REV_00\4&25be462b&0&0008
ElementName	string	HCA 0
EnabledDefault	uint16	2
EnabledState	uint16	5
ErrorDescription	string	OK
HealthState	uint16	5
InstanceID	string	PCI\VEN_15B3&DEV_1003&SUBSYS_005015B3&REV_00\4&25be462b&0&0008
LastErrorCode	uint32	0
Name	string	HCA 0
PrimaryStatus	uint16	1
RequestedState	uint16	12
Status	string	OK
SystemCreationClassName	string	CIM_System
SystemName	string	WINDOWS-124SFT7
TransitioningToState	uint16	12
PartNumber	uint16[]	MCX354A-FCBT
FirmwareVersion	uint16[]	2.11.500
DriverVersion	uint16[]	4.41.14370.0
BusType	string[]	PCI-E 5.0 Gbps x4
OperationalStatus	uint16[]	{2}
StatusDescriptions	string[]	{OK}

5.4.1.2 Class Methods

5.4.1.2.1 RequestStateChange (uint32)

Table 18: RequestStateChange Method Parameters

Name	Qualifier	Type
RequestedState	IN	uint16
TimeoutPeriod	IN	datetime
Job	OUT	REF CIM_ConcreteJob

5.4.2 MLNX_SystemDevice – Extend CIM_SystemDevice

5.4.2.1 Associations

Table 19: MLNX_SystemDevice Associations

Name	Reference Class
GroupComponent	CIM_System
PartComponent	CIM_LogicalDevice

5.4.3 MLNX_PCIDeviceCapabilities – Extend CIM_Capabilities

5.4.3.1 Class Properties

Table 20: MLNX_PCIDeviceCapabilities Class Properties

Name	Data Type	Expected Sample
Caption	string	MLNX_PCIDeviceCapabilities 'Mellanox ConnectX-3 (MT04099) Network Adapter'
Description	string	Mellanox ConnectX-3 (MT04099) Network Adapter Capabilities
ElementName	string	HCA 0
InstanceID	string	PCI\VEN_15B3&DEV_6732&SUBSYS_001315B3&REV_B0\4&d366597&0&0018
Name	string	HCA 0
SystemName	string	WINDOWS-124SFT7
DualPort	boolean	True
PortOneIb	boolean	True
PortTwoIb	boolean	True
PortOneEth	boolean	True
PortTwoEth	boolean	True
PortOneDoSenseAllowed	boolean	False
PortTwoDoSenseAllowed	boolean	False
PortOneAutoSense	boolean	True
PortTwoAutoSense	boolean	True

Name	Data Type	Expected Sample
PortOneDefault	boolean	False
PortTwoDefault	boolean	False

5.4.4 MLNX_PCIDeviceSettingData – Extend CIM_SettingData

5.4.4.1 Class Properties

Table 21: MLNX_PCIDeviceSettingData Class Properties

Name	Data Type	Expected Sample
ElementName	string	HCA 1
InstanceID	string	PCI\VEN_15B3&DEV_1003&SUBSYS_005015B3&REV_00\4&25be462b&0&0008
Caption	string	MLNX_PCIDeviceSettingData 'Mellanox ConnectX (MT26418) - PCIe 2.0 5GT/s, IB DDR / 10GigE Network Adapter'
Description	string	Mellanox ConnectX (MT26418) - PCIe 2.0 5GT/s, IB DDR / 10GigE Network Adapter
Name	string	HCA 1
Source	uint32	3
SystemName	string	WINDOWS-124SFT7

5.4.5 MLNX_PCIDeviceSriovSettingData – Extend MLNX_PCIDeviceSettingData

5.4.5.1 Class properties

Table 22: MLNX_PCIDeviceSriovSettingData Class Properties

Name	Data Type	Expected Sample
ElementName	string	HCA 1
InstanceID	string	PCI\VEN_15B3&DEV_1003&SUBSYS_005015B3&REV_00\4&25be462b&0&0008
Caption	string	MLNX_PCIDeviceSettingData 'Mellanox ConnectX (MT26418) - PCIe 2.0 5GT/s, IB DDR / 10GigE Network Adapter'
Description	string	Mellanox ConnectX (MT26418) - PCIe 2.0 5GT/s, IB DDR / 10GigE Network Adapter
Name	string	HCA 1
Source	uint32	3
SystemName	string	WINDOWS-124SFT7
SriovEnable	boolean	true
SriovPortMode	uint32	0
SriovPort1NumVFs	uint32	16

Name	Data Type	Expected Sample
SriovPort2NumVFs	uint32	0

5.4.5.2 Class Methods

5.4.5.2.1 Set Value

Table 23: Set Value Method Parameters

Name	Qualifier	Type
SriovEnable	IN	boolean
SriovPortMode	IN	uint32
SriovPort1NumVFs	IN	uint32
SriovPort2NumVFs	IN	uint32

5.4.6 MLNX_PCIDevicePortTypeSettingData – Extend MLNX_PCIDeviceSettingData

5.4.6.1 Class Properties

Table 24: MLNX_PCIDevicePortTypeSettingData Class Properties

Name	Data Type	Expected Sample
ElementName	string	HCA 1
InstanceID	string	PCI\VEN_15B3&DEV_1003&SUBSYS_005015B3&REV_00\4&25be462b&0&0008
Caption	string	MLNX_PCIDeviceSettingData 'Mellanox ConnectX (MT26418) - PCIe 2.0 5GT/s, IB DDR / 10GigE Network Adapter'
Description	string	Mellanox ConnectX (MT26418) - PCIe 2.0 5GT/s, IB DDR / 10GigE Network Adapter
Name	string	HCA 1
Source	uint32	3
SystemName	string	WINDOWS-124SFT7
Configuration	string[]	{eth, ib}
CurrentSetting	string[]	{eth, ib}
DefaultSetting	string[]	{ib, ib}

5.4.6.2 Class Methods

5.4.6.2.1 Set Value (uint32)

Table 25: Set Value Method Parameters

Name	Qualifier	Type
port1	IN	string
port2	IN	string

5.4.7 MLNX_IBPort – Extend CIM_IBPort

5.4.7.1 Class Properties

Table 26: MLNX_IBPort Class Properties

Name	Data Type	Expected Sample
LIDMask	uint8	0
LinkSpeedActive	uint8	2
LinkWidthActive	uint16	1
MaxMsgSize	uint64	1073741824
MaxVls	uint16	2
NumGids	uint16	128
NumPkeys	uint16	128
QkeyCtr	uint16	0
SmSI	uint8	0
SubnetTimeout	uint8	0
Transport	string	IB
PortType	uint16	62
ActiveMaximumTransmissionUnit	uint64	4096
AutoSense	boolean	True
Caption	string	MLNX_IBPort 'Mellanox HCA 1 RDMA Port 2'
CreationClassName	string	MLNX_IBPort
Description	string	Mellanox HCA 1 RDMA Port 2
DeviceID	string	0002c90300454472
ElementName	string	HCA 1 RDMA Port 2
EnabledDefault	uint16	2
EnabledState	uint16	3
FullDuplex	boolean	True
InstanceID	string	0002c90300454472
LinkTechnology	uint16	3
MaxSpeed	uint64	0
Name	string	HCA 1 RDMA Port 2
NetworkAddresses	String[]	{0000}
PermanentAddress	string	0002c90300454472
PortNumber	uint16	2
PciLocation	string	1:0:0
RequestedState	uint16	12

Name	Data Type	Expected Sample
Speed	uint64	0
SupportedMaximumTransmissionUnit	uint64	4096
SystemCreationClassName	string	CIM_System
SystemName	string	WINDOWS-124SFT7
TransitioningToState	uint16	12
OperationalStatus	uint16[]	{6}
StatusDescriptions	string[]	{PORT_DOWN}

5.4.8 MLNX_IBPortCounters – Extend CIM_IBPortStatistics

5.4.8.1 Class Properties

Table 27: MLNX_IBPortCounters Class Properties

Name	Data Type	Expected Sample
ElementName	string	HCA 1 RDMA Port 2
InstanceID	string	0002c9030008eb11
Caption	string	MLNX_IBPort 'Mellanox HCA 1 RDMA Port 1'
Description	string	Mellanox HCA 1 RDMA Port 1
SystemName	string	WINDOWS-124SFT7
SampleInterval		00:00:00
StartStatisticTime		
StatisticTime		7/18/2013 3:26:46 AM
BytesReceived		0
BytesTransmitted		0
PacketsReceived		0
PacketsTransmitted		0
ExcessiveBufferOverrunErrors		0
LinkDownedCounter		0
LinkErrorRecoveryCounter		0
LocalLinkIntegrityErrors		0
PortRcvConstraintErrors		0
PortRcvErrors		0
PortRcvRemotePhysicalErrors		0
PortRcvSwitchRelayErrors		0
PortXmitConstraintErrors		0
PortXmitDiscards		0
SymbolErrorCounter		0

Name	Data Type	Expected Sample
VL15Dropped		0

5.4.9 MLNX_IBPortControlledBy – Extend CIM_ ControlledBy

5.4.9.1 Associations

Table 28: MLNX_IBPortControlledBy Associations

Name	Reference Class
Antecedent	CIM_Controller
Dependent	CIM_LogicalDevice

5.5 Ethernet Port Profile

The Ethernet Port Profile adding the capability to represent an Ethernet port, its associated controller, and Ethernet interfaces. Associations with the port's physical aspects and profile-implementation version information are also modeled in this profile. This profile specialize the Host LAN Network Port Profile. CIM_EthernetPort is central class of this profile. The CIM_EthernetPort class represents the Ethernet port. The CIM_LANEndpoint class represents an access point at the data-link layer, identified by a MAC address to which the Ethernet port will respond on the network.

5.5.1 MLNX_NetAdapter – Extend CIM_ EthernetPort

5.5.1.1 Class Properties

Table 29: MLNX_NetAdapter Class Properties

Name	Data Type	Expected Sample
ActiveMaximumTransmissionUnit	uint64	4092
AutoSense	boolean	True
Availability	uint16	9
Caption	string	MLNX_EthernetPort 'Mellanox ConnectX-2 IPoIB Adapter #2'
CreationClassName	string	MLNX_NetAdapter
Description	string	Mellanox ConnectX-2 IPoIB Adapter #2
DeviceID	string	36
ElementName	string	Ethernet
EnabledDefault	uint16	2
EnabledState	uint16	5
ErrorDescription	string	IfOperStatusDown
FullDuplex	boolean	True
HealthState	uint16	15

Name	Data Type	Expected Sample
IdentifyingDescriptions	string[]	{vNIC}
InstanceID	string	{051316FB-0308-4283-94EB-C43A4E03A407}
LastErrorCode	uint32	0
LinkTechnology	uint16	3
MaxSpeed	uint64	40000000000
Name	string	Ethernet 6
NetworkAddresses	String[]	{F79F95ACF79F95A}
OtherIdentifyingInfo		{0}
PciLocation	string	2:0:0
PermanentAddress	string	F79F95ACF79F95A
PortNumber	uint16	2
PrimaryStatus	uint16	3
RequestedState	uint16	12
Speed	uint64	40000000000
Status	string	IfOperStatusDown
SystemCreationClassName	string	CIM_System
SystemName	string	WINDOWS-124SFT7
TransitioningToState	uint16	12
OperationalStatus	uint16[]	{12}
StatusDescriptions	string[]	{Down}

5.5.1.2 Class Methods

Table 30: MLNX_NetAdapter Class Methods

Name	Qualifier	Type
RequestedState	IN	uint16
TimeoutPeriod	IN	datetime
Job	OUT	REF CIM_ConcreteJob

5.5.2 MLNX_NetAdapterStatistics – Extend CIM_NetworkPortStatistics

5.5.2.1 MLNX_NetAdapterGenStatistics - Extend MLNX_NetAdapterStatistics

5.5.2.1.1 Class Properties

Table 31: MLNX_NetAdapterGenStatistics Class Properties

Name	Data Type	Expected Sample
Caption	String	MLNX_NetAdapterGenStatistics 'Mellanox

Name	Data Type	Expected Sample
		ConnectX-3 Ethernet Adapter'
Description	String	Mellanox ConnectX-3 Ethernet Adapter
ElementName	String	Ethernet 30
InstanceID	String	{FC9A6B3D-8A4A-49BB-9DA7-AF551AF033D8 }
StartStatisticTime	Datetime	
StatisticTime	Datetime	
SampleInterval	Datetime	00:00:00
BytesTransmitted	UInt64	57905
BytesReceived	UInt64	1298
PacketsTransmitted	UInt64	
PacketsReceived	UInt64	7
ifInDiscards	UInt64	657
ifHCInBroadcastOctets	UInt64	692
ifHCInBroadcastPkts	UInt64	2
ifHCInMulticastOctets	UInt64	90
ifHCInMulticastPkts	UInt64	1
ifHCInOctets	UInt64	1298
ifHCInUcastOctets	UInt64	516
ifHCInUcastPkts	UInt64	4
ifHCOutBroadcastOctets	UInt64	26075
ifHCOutBroadcastPkts	UInt64	263
ifHCOutMulticastOctets	UInt64	31560
ifHCOutMulticastPkts	UInt64	391
ifHCOutOctets	UInt64	57905
ifHCOutUcastOctets	UInt64	270
ifHCOutUcastPkts	UInt64	3
ifHCInCrcError	UInt64	0
ifHCInCrcOverRun	UInt64	0
ifInDiscards	UInt64	0
ifInErrors	UInt64	0
ifOutDiscards	UInt64	0
ifOutErrors	UInt64	0



NOTE: Extending CIM_SettingData for MLNX_NetAdapterSettingData for MLNX_NetAdapterRoceSettingData applies only to WinOF 4.40 or WinOF 4.80 and above. Running this profile in version lower than WinOF 4.80 will result in error message

5.5.2.1.2 Class Properties

Table 32: MLNX_NetAdapterRoceSettingData Class Properties

Name	Data Type	Expected Sample
Caption	string	MLNX_NetAdapterRoceSettingData Mellanox ConnectX-2 Ethernet Adapter
Description	string	Mellanox ConnectX-2 Ethernet Adapter
ElementName	string	Ethernet 7
InstanceID	string	{317AF5E4-DC0D-4679-A358-9AC959D96440}
InterfaceDescription	string	Mellanox ConnectX-2 Ethernet Adapter
Name	string	Ethernet 7
Source	uint32	3
SystemName	string	WINDOWS-124SFT7
Enabled	boolean	True
PortNumber	uint16	1
RoceMode	string	1.0

5.5.3 MLNX_NetAdapterControlledBy – Extend CIM_ControlledBy

5.5.3.1 Associations

Table 33: MLNX_NetAdapterControlledBy Associations

Name	Reference Class
Antecedent	CIM_Controller
Dependent	CIM_LogicalDevice

5.5.4 MLNX_NetAdapterFlowControlSettingData extend CIM_SettingData

5.5.4.1 Class Properties

Table 34: MLNX_NetAdapterFlowControlSettingData Class Properties¹

Name	Data Type	Expected Sample
Caption	string	MLNX_NetAdapterFlowControlSettingData Mellanox ConnectX-2 Ethernet Adapter
Description	string	Mellanox ConnectX-2 Ethernet Adapter

¹ Only for connectX@-3 and requires firmware version 2.3.1100 and above.

Name	Data Type	Expected Sample
ElementName	string	Ethernet 7
InstanceID	string	{317AF5E4-DC0D-4679-A358-9AC959D96440}
InterfaceDescription	string	Mellanox ConnectX-2 Ethernet Adapter
Name	string	Ethernet 7
Source	uint32	2
SystemName	string	WINDOWS-124SFT7
TxUntagPriorityTag	uint8	0
RxUntaggedMapToLossless	uint8	0
RroceDscpMarkPriorityFlowControl_0	uint8	0
RroceDscpMarkPriorityFlowControl_1	uint8	1
RroceDscpMarkPriorityFlowControl_2	uint8	2
RroceDscpMarkPriorityFlowControl_3	uint8	3
RroceDscpMarkPriorityFlowControl_4	uint8	4
RroceDscpMarkPriorityFlowControl_5	uint8	5
RroceDscpMarkPriorityFlowControl_6	uint8	6
RroceDscpMarkPriorityFlowControl_7	uint8	7
PFCTx	string	off
PF CRx	string	global

5.6 Physical Asset Profile

The Physical Asset Profile adding the capability to describe the physical aspects of logical elements that the implementation is instantiating. This profile also describes the relationship between the physical elements and the profile's registration for the schema implementation and version information.

5.6.1 MLNX_Card – Extend CIM_Card

5.6.1.1 Class Properties

Table 35: MLNX_CardClass Properties

Name	Data Type	Expected Sample
SlotLayout	string	2:0:0
Caption	string	MLNX_Card 'Device 26418 with PSID MT_0D20110008'
CreationClassName	string	MLNX_Card
Description	string	Card
ElementName	string	mt26418_pci_cr0
InstanceID	string	mt26418_pci_cr0

Name	Data Type	Expected Sample
Manufacturer	string	MLNX
Model	string	26418
Name	string	MLNX VPI Adapter MT1008X01108
PartNumber	string	MHGH29B-XTR
Removable	boolean	True
RemovalConditions	uint16	2
Replaceable	boolean	True
SerialNumber	string	MT1008X01108
SKU	string	MHGH29B-XTR
Tag	string	2:0:0
Version	string	MT_OD20110008
PackageType	uint16	9

5.6.2 MLNX_Realizes – Extend CIM_Realizes

5.6.2.1 Associations

Table 36: MLNX_Realizes Associations

Name	Reference Class
Antecedent	MLNX_Card
Dependent	MLNX_PCIDevice

5.7 Host LAN Network Port Profile

The Host LAN Network Port Profile adds the capability to represent a network port that provides a LAN interface in a managed host system. This profile includes a specification of the network port, its associated controller, associated network interfaces, and the realization of the connection in a physical connector.

The CIM_NetworkPort is the Central Class of this profile and instance of CIM_NetworkPort is the Central Instance of this profile. The CIM_NetworkPort class represents a network port of the system with one or more communication interface represented through CIM_LANEndpoint. A given CIM_LANEndpoint on the network port is identified by a MAC address to which the network port will respond. A network port can have an associated controller. The controller is represented by an instance of CIM_PortController. The relationship between the controller and port is modeled through the CIM_ControlledBy association.

5.8 Diagnostics CDMv2 Profile

Diagnostics (CDMv2) profile purpose is to ensure interoperability in the use of Web-Based Enterprise Management (WBEM) services for a diagnostics.

The goal of the Common Diagnostic Model (CDM) is to define industry-standard building blocks, based on and consistent with the DMTF CIM, that enables seamless integration of vendor-supplied diagnostic services into system and SAN management frameworks.

The CDM is an architecture and methodology for exposing system diagnostic instrumentation through the CIM standard interfaces.

A primary objective of the CDM is to standardize the interfaces that diagnostic developers create for their OS-Present Diagnostics in the operating environment, making the diagnostics accessible to all applications that query CIM for diagnostic data or register with CIM to execute diagnostic methods and receive results.

The currently supported diagnostics are implemented on the driver level. There is no option to change the settings, but the diagnostic framework is implemented to enable adding external diagnostics tools and scripts. The log files are saved at the provider root path. Available Tests:

- TEST_CONTROL_REGISTERS - verifies access to MAC/GUID layer registers.
- TEST_INTERRUPT - verifies that the HCA properly raises interrupt signals to the host.
- TEST_INTERNAL_LOOPBACK - verifies the internal traffic path for received (Rx) and transmitted (Tx) data.
- TEST_LINK - verifies the HCA port link integrity.
- TEST_SPEED - verifies the port speed according to the HCA specification.

5.8.1 MLNX_DiagnosticTestHca – Extend MLNX_DiagnosticTest

5.8.1.1 Class Properties

Table 37: MLNX_DiagnosticTestHca Class Properties

Name	Data Type	Expected Sample
Caption	string	MLNX_DiagnosticTest 'Diagnostic Speed Test.'
CreationClassName	string	MLNX_DiagnosticTestHca
Description	string	This test verifies the port speed according to the HCA specification.
ElementName	string	TEST_SPEED
InstanceID	string	MLNX:TEST_SPEED
Name	string	TEST_SPEED
EnabledDefault	uint16	2
EnabledState	uint16	5
RequestedState	uint16	12
TransitioningToState	uint16	12
PrimaryOwnerContact	string	http://www.mellanox.com
PrimaryOwnerName	string	Mellanox Technologies
SystemCreationClassName	string	CIM_System

Name	Data Type	Expected Sample
SystemName	string	WINDOWS-124SFT7
Characteristics	uint16[]	{8}
ResourcesUsed	uint16[]	{7, 12}
TestTypes	uint16[]	{2, 4}

5.8.1.2 Class Methods

5.8.1.2.1 RunDiagnostic (uint32)

Table 38: RunDiagnostic Method Parameters

Name	Qualifier	Type
DiagSetting	IN	REF CIM_DiagnosticSetting
Job	OUT	REF CIM_ConcreteJob
JobSetting	IN	REF CIM_JobSettingData
ManagedElement	IN	REF CIM_ManagedElement

5.8.1.2.2 RunDiagnosticService (uint32)

Table 39: RunDiagnosticService Method Parameters

Name	Qualifier	Type
DiagnosticSetting	IN	string
JobSetting	IN	string
Job	OUT	REF CIM_ConcreteJob
ManagedElement	IN	REF CIM_ManagedElement

5.8.2 MLNX_DiagnosticConcreteJob – Extend CIM_ConcreteJob

5.8.2.1 Class Properties

Table 40: MLNX_DiagnosticConcreteJob Class Properties

Name	Data Type	Expected Sample
Caption	string	MLNX_DiagnosticConcreteJob 'Diagnostic Speed Job.'
Description	string	Diagnostic Speed Job.
ElementName	string	TEST_SPEED
InstanceID	string	MLNX: TEST_SPEED
Name	string	TEST_SPEED
JobRunTimes	uint16	
JobState	uint16	2
TimeBeforeRemoval	datetime	00:05:00
JobStatus	uint16	

Name	Data Type	Expected Sample
StartTime	string	
ErrorCode	string	Mellanox Technologies

5.8.3 MLNX_DiagnosticLog – Extend CIM_DiagnosticJob

5.8.3.1 Class Properties

Table 41: Extend CIM_DiagnosticJob Class Properties

Name	Data Type	Expected Sample
Caption	string	MLNX_DiagnosticLog 'Log file for DiagnosticTestHca'
Description	string	Log file for DiagnosticTestHca
ElementName	string	DiagnosticTestHca.log
InstanceID	string	C:\Program Files\Mellanox\MLNX_CIMProvider\DiagnosticTestHca.log
Name	string	DiagnosticTestHca.log
CurrentNumberOfRecords	UInt64	0
LogState	uint16	4
MaxNumberOfRecords	UInt64	
OverwritePolicy	uint16	0
EnabledDefault	uint16	2
EnabledState	uint16	5
RequestedState	uint16	12
TransitioningToState	uint16	12

5.8.4 MLNX_AvailableDiagnosticService – Extend CIM_AvailableDiagnosticService

5.8.4.1 Associations

Table 42: MLNX_AvailableDiagnosticService Associations

Name	Reference Class
ServiceProvided	CIM_DiagnosticService
UserOfService	CIM_ManagedElement

5.8.5 MLNX_DiagnosticUseOfLog – Extend CIM_UseOfLog

5.8.5.1 Associations

Table 43: MLNX_DiagnosticUseOfLog Associations

Name	Reference Class
------	-----------------

Name	Reference Class
Antecedent	MLNX_DiagnosticLog
Dependent	MLNX_DiagnosticTest

5.9 Indications Profile

The Indications Profile defines the CIM elements that are used to subscribe for indications of unsolicited events. It enables a server-side implementation to advertise the possible indications. The Indications Profile defines the content of indications from autonomous and component profiles implemented by CIM-based management instrumentation. The Indications Profile describes the necessary properties and methods to describe the indications supported by managed elements and how a client subscribes to those indications.

5.10 Drivers Services Profile

For further information on driver options please refer to WinOF User Manual.

5.10.1 MLNX_DriverService – Extend CIM_Service

5.10.1.1 Class Properties

Table 44: MLNX_DriverService Class Properties

Name	Data Type	Expected Sample
Caption	string	MLNX_DriverService ‘Mellanox ConnectX based Ethernet Adapter (NDIS 6.3) Driver’
Description	string	Mellanox ConnectX based Ethernet Adapter (NDIS 6.3) Driver
ElementName	string	mlx4eth63
InstanceID	string	mlx4eth63
Name	string	mlx4eth63
Status	string	OK
CreationClassName	string	MLNX_DriverService
PrimaryOwnerContact	string	http://www.mellanox.com/page/support_index
PrimaryOwnerName	string	Mellanox Technologies
Started	boolean	True
StartMode	string	Service boot start
SystemCreationClassName	string	CIM_System
SystemName	string	WINDOWS-124SFT7
EnabledDefault	uint16	2
EnabledState	uint16	5
RequestedState	uint16	12
TransitioningToState	uint16	12

5.10.2 MLNX_DriverCoreCapabilities – Extend MLNX_DriverCapabilities

5.10.2.1 Class Properties

Table 45: MLNX_DriverCoreCapabilities Class Properties

Name	Data Type	Expected Sample
Caption	string	MLNX_DriverCoreCapabilities 'Mellanox ConnectX Bus Enumerator Capabilities'
Description	string	Mellanox ConnectX Bus Enumerator Capabilities
ElementName	string	mlx4_bus
InstanceID	string	mlx4_bus
Name	string	mlx4_bus
SystemName	string	WINDOWS-124SFT7
Set4kMtuMin	uint32	0
Set4kMtuMax	uint32	1
QosMin	uint32	0
QosMax	uint32	1
BlockMcastLoopBackMin	uint32	
BlockMcastLoopBackMax	uint32	
UsePrioMin	uint32	0
UsePrioMax	uint32	1
InterruptFromFirstPacketMin	uint32	0
InterruptFromFirstPacketMax	uint32	1
LogNumQpMin	uint32	17
LogNumQpMax	uint32	24
LogNumRdmaRcMin	uint32	0
LogNumRdmaRcMax	uint32	7
LogNumSrQMin	uint32	8
LogNumSrQMax	uint32	23
LogNumCqMin	uint32	8
LogNumCqMax	uint32	24
LogNumMcgMin	uint32	8
LogNumMcgMax	uint32	16
LogNumMptMin	uint32	17
LogNumMptMax	uint32	31
LogNumMttMin	uint32	20
LogNumMttMax	uint32	31
LogNumMacMin	uint32	0

Name	Data Type	Expected Sample
LogNumMacMax	uint32	7
LogNumVlanMin	uint32	0
LogNumVlanMax	uint32	7
NumFcExchMin	uint32	0
NumFcExchMax	uint32	32768
LogMtlsPerSegMin	uint32	1
LogMtlsPerSegMax	uint32	5
ModeFlagsMin	uint32	0
ModeFlagsMax	uint32	-1
StatFlagsMin	uint32	0
StatFlagsMax	uint32	-1
SingleMsixNumMin	uint32	0
SingleMsixNumMax	uint32	-1
MultiMsixNumMin	uint32	0
MultiMsixNumMax	uint32	-1
SingleEqNumMin	uint32	0
SingleEqNumMax	uint32	-1
MultiEqNumMin	uint32	0
MultiEqNumMax	uint32	-1
MaxContQuantMin	uint32	0
MaxContQuantMax	uint32	-1
DebugFlagsMin	uint32	0
DebugFlagsMax	uint32	65535
DebugLevelMin	uint32	
DebugLevelMax	uint32	
MaximumWorkingThreadsMin	uint32	1
MaximumWorkingThreadsMax	uint32	256

5.10.3 MLNX_DriverEthCapabilities – Extend MLNX_DriverCapabilities

5.10.3.1 Class Properties

Table 46: MLNX_DriverEthCapabilities Class Properties

Name	Data Type	Expected Sample
Caption	string	MLNX_DriverCoreCapabilities 'Mellanox ConnectX Bus Enumerator Capabilities'
Description	string	Mellanox ConnectX Bus Enumerator Capabilities
ElementName	string	mlx4_bus

Name	Data Type	Expected Sample
InstanceID	string	mlx4_bus
Name	string	mlx4_bus
SystemName	string	WINDOWS-124SFT7
ModeFlagsMin	uint32	0
ModeFlagsMax	uint32	-1
NdkDebugFlagsMin	uint32	0
NdkDebugFlagsMax	uint32	-1
NdkDebugLevelMin	uint32	0
NdkDebugLevelMax	uint32	5

5.10.4 MLNX_DriverIbCapabilities – Extend MLNX_DriverCapabilities

5.10.4.1 Class Properties

Table 47: MLNX_DriverIbCapabilities Class Properties

Name	Data Type	Expected Sample
Caption	string	MLNX_DriverCoreCapabilities 'Mellanox ConnectX Bus Enumerator Capabilities'
Description	string	Mellanox ConnectX Bus Enumerator Capabilities
ElementName	string	mlx4_bus
InstanceID	string	mlx4_bus
Name	string	mlx4_bus
SystemName	string	WINDOWS-124SFT7
DebugFlagsMin	uint32	0
DebugFlagsMax	uint32	65535
IbalDebugFlagsMin	uint32	0
IbalDebugFlagsMax	uint32	255
IbalDebugLevelMin	uint32	0
IbalDebugLevelMax	uint32	5

5.10.5 MLNX_DriverIpoVerIbCapabilities – Extend MLNX_DriverCapabilities

5.10.5.1 Class Properties

Table 48: MLNX_DriverIpoVerIbCapabilities Class Properties

Name	Data Type	Expected Sample
Caption	string	MLNX_DriverCoreCapabilities 'Mellanox ConnectX Bus Enumerator Capabilities'
Description	string	Mellanox ConnectX Bus Enumerator Capabilities
ElementName	string	mlx4_bus

Name	Data Type	Expected Sample
InstanceID	string	mlx4_bus
Name	string	mlx4_bus
SystemName	string	WINDOWS-124SFT7
ModeFlagsMin	uint32	0
ModeFlagsMax	uint32	-1
NdkDebugFlagsMin	uint32	0
NdkDebugFlagsMax	uint32	-1
NdkDebugLevelMin	uint32	0
NdkDebugLevelMax	uint32	5

5.10.6 MLNX_DriverCoreSettingData – Extend MLNX_DriverSettingData

5.10.6.1 Class Properties

Table 49: MLNX_DriverCoreSettingData Class Properties

Name	Data Type	Expected Sample
Caption	string	DriverCoreSettingData 'mlx4_bus'
Description	string	Mellanox Core Driver Option Settings
ElementName	string	mlx4_bus
InstanceID	string	mlx4_bus
Name	string	mlx4_bus
Source	uint32	3
SystemName	string	WINDOWS-124SFT7
Set4kMtu	boolean	True
Qos	boolean	False
BlockMcastLoopBack	boolean	False
UsePrio	boolean	False
InterruptFromFirstPacket	boolean	True
LogNumQp	uint32	17
LogNumRdmaRc	uint32	4
LogNumSrq	uint32	16
LogNumCq	uint32	16
LogNumMcg	uint32	13
LogNumMpt	uint32	17
LogNumMtt	uint32	20
LogNumMac	uint32	7
LogNumVlan	uint32	7

Name	Data Type	Expected Sample
NumFcExch	uint32	0
LogMttPerSeg	uint32	3
ModeFlags	uint32	0
StatFlags	uint32	0
SingleMsixNum	uint32	0
MultiMsixNum	uint32	0
SingleEqNum	uint32	0
MultiEqNum	uint32	0
MaxContQuant	uint32	0
DebugFlags	uint32	65535
DebugLevel	uint32	3
MaximumWorkingThreads	uint32	4
RoceMode	string	1.0
SriovEnable	boolean	False
SriovPort1NumVFs	uint32	16
SriovPort2NumVFs	uint32	0
SriovPortMode		

5.10.6.2 Class Methods

5.10.6.2.1 SetValue (uint32)

Table 50: SetValue Method Parameters

Name	Qualifier	Type
LogNumRdmaRc	IN	uint32
LogNumSrq	IN	uint32
LogNumCq	IN	uint32
LogNumMcg	IN	uint32
LogNumMpt	IN	uint32
LogNumMtt	IN	uint32
LogNumMac	IN	uint32
LogNumVlan	IN	uint32
LogMttPerSeg	IN	uint32
ModeFlags	IN	uint32
DebugFlags	IN	uint32
DebugLevel	IN	uint32
MaximumWorkingThreads	IN	uint32
RoceMode	IN	String

Name	Qualifier	Type
Set4kMtu	IN	boolean
SriovEnable	IN	boolean
SriovPortMode	IN	uint32
SriovPort1NumVFs	IN	uint32
SriovPort2NumVFs	IN	uint32

5.10.7 MLNX_DriverEthSettingData – Extend MLNX_DriverSettingData

5.10.7.1 Class Properties

Table 51: MLNX_DriverEthSettingData Class Properties

Name	Data Type	Expected Sample
Caption	string	DriverEthSettingData 'mlx4eth63'
Description	string	Mellanox mlx4eth63 Option Settings
ElementName	string	mlx4eth63
InstanceID	string	mlx4eth63
Name	string	mlx4eth63
Source	uint32	3
SystemName	string	WINDOWS-124SFT7
ModeFlags	uint32	0
NdkDebugFlags	uint32	4095
NdkDebugLevel	uint32	2

5.10.7.2 Class Methods

5.10.7.2.1 SetValue (uint32)

Table 52: SetValue Method Parameters

Name	Qualifier	Type
ModeFlags	IN	uint32
NdkDebugFlags	IN	uint32
NdkDebugLevel	IN	uint32

5.10.8 MLNX_DriverIbSettingData – Extend MLNX_DriverSettingData

5.10.8.1 Class Properties

Table 53: MLNX_DriverIbSettingData Class Properties

Name	Data Type	Expected Sample
Caption	string	DriverIbSettingData 'ibbus'
Description	string	Mellanox ibbus Option Settings

Name	Data Type	Expected Sample
ElementName	string	ibbus
InstanceID	string	ibbus
Name	string	ibbus
Source	uint32	3
SystemName	string	WINDOWS-124SFT7
DebugFlags	uint32	2147483648
IbalDebugFlags	uint32	16777215
IbalDebugLevel	uint32	2

5.10.8.2 Class Methods

5.10.8.2.1 SetValue (uint32)

Table 54: SetValue Method Parameters

Name	Qualifier	Type
DebugFlags	IN	uint32
IbalDebugFlags	IN	uint32
IbalDebugLevel	IN	uint32

5.10.9 MLNX_DriverIpOverIbSettingData – Extend MLNX_DriverSettingData

5.10.9.1 Class Properties

Table 55: MLNX_DriverIpOverIbSettingData Class Properties

Name	Data Type	Expected Sample
Caption	string	DriverIpOverIbSettingData 'ipoib6x'
Description	string	Mellanox ipoib6x Option Settings
ElementName	string	ipoib6x
InstanceID	string	ipoib6x
Name	string	ipoib6x
Source	uint32	3
SystemName	string	WINDOWS-124SFT7
ModeFlags	uint32	0
NdkDebugFlags	uint32	4095
NdkDebugLevel	uint32	2

5.10.9.2 Class Methods

5.10.9.2.1 SetValue (uint32)

Table 56: SetValue Method Parameters

Name	Qualifier	Type
------	-----------	------

Name	Qualifier	Type
ModeFlags	IN	uint32
NdkDebugFlags	IN	uint32
NdkDebugLevel	IN	uint32

5.10.10 MLNX_DriverElementCapabilities – Extend CIM_ElementCapabilities

5.10.10.1 Associations

Table 57: MLNX_DriverElementCapabilities Associations

Name	Reference Class
ManagedElement	CIM_ManagedElement
Capabilities	CIM_Capabilities

5.10.11 MLNX_DriverElementSettingData – Extend CIM_ElementSettingData

5.10.11.1 Associations

Table 58: MLNX_DriverElementSettingData Associations

Name	Reference Class
ManagedElement	CIM_ManagedElement
SettingData	CIM_SettingData