SOFTWARE MANUAL

Mellanox MLNX-OS® XML API Reference Guide for VPI

Rev 1.6

Software Version 3.6.8008

NOTE:
THIS HARDWARE, SOFTWARE OR TEST SUITE PRODUCT ("PRODUCT(S)") AND ITS RELATED DOCUMENTATION ARE PROVIDED BY MELLANOX TECHNOLOGIES "AS IS" WITH ALL FAULTS OF ANY KIND AND SOLELY FOR THE PURPOSE OF AIDING THE CUSTOMER IN TESTING APPLICATIONS THAT USE THE PRODUCTS IN DESIGNATED SOLUTIONS. THE CUSTOMER'S MANUFACTURING TEST ENVIRONMENT HAS NOT MET THE STANDARDS SET BY MELLANOX TECHNOLOGIES TO FULLY QUALIFY THE PRODUCT(S) AND/OR THE SYSTEM USING IT. THEREFORE, MELLANOX TECHNOLOGIES CANNOT AND DOES NOT GUARANTEE OR WARRANT THAT THE PRODUCTS WILL OPERATE WITH THE HIGHEST QUALITY. ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT ARE DISCLAIMED. IN NO EVENT SHALL MELLANOX BE LIABLE TO CUSTOMER OR ANY THIRD PARTIES FOR ANY DIRECT, INDIRECT, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES OF ANY KIND (INCLUDING, BUT NOT LIMITED TO, PAYMENT FOR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY FROM THE USE OF THE PRODUCT(S) AND RELATED DOCUMENTATION EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

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

Table of Contents

Document Revision History

Rev 1.6 – June 30, 2017

Updated:

- “Transport Layer” on page 11

Rev 1.5 – May 5, 2016

- N/A

Rev 1.4 – September 30, 2016

Added:

- “/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/priorities/*” on page 167

Rev 1.3 – June 02, 2016

Updated:

- See “Query Request via HTTP GET Transport Layer” on page 24

Rev 1.2 – May 21, 2016

Updated:

- “/mlnxos/v1/vsr/<STRING>/dcb/pfc/enabled” on page 136
- “/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/priorities/*” on page 167
- “/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/enabled” on page 144
- “/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/operational_state” on page 146
- “/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/supported_speed” on page 147
- “/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/actual_speed” on page 148
- “/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/configured_speed” on page 149
- “/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/actual_rate” on page 150
- “/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/mtu” on page 151
- “/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/supported_mtu” on page 152
- “/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/supported_width” on page 176
- “/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/actual_width” on page 177
- “/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/configured_width” on page 178

Added:

- “/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/priorities/<UINT8>/traffic_class” on page 168

Rev 1.1 – January 31, 2016

Added:

- “/mlnxos/v1/chassis/mgmt_interfaces/<IFNAME>/physical_address” on page 67
- “/mlnxos/v1/vsr/<STRING>/dcb/pfc/enabled” on page 136
- “/mlnxos/v1/vsr/<STRING>/dcb/pfc/priorities/*” on page 137
- “/mlnxos/v1/vsr/<STRING>/dcb/pfc/priorities/<UINT8>/enabled” on page 138
- “/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/dcb/pfc/admin_mode” on page 159
- “/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/priorities/*” on page 167

Rev 1.0 – August 24, 2015

First release

About this Manual

This guide provides general information concerning MLNX-OS® XML API.

Intended Audience

This guide is intended for application developers wishing to integrate with the switch through a programmatic API for Mellanox Technologies' MLNX-OS switch platforms.

Related Documentation

The following documents can be found on the Mellanox Support webpage (support.mellanox.com).

Table 1 - Reference Documents

Document Name	Description
Mellanox MLNX-OS® Software User Manual	This document contains information regarding configuring and managing Mellanox Technologies' SwitchX® Switch Platforms as well as description of all MLNX-OS CLI commands.

Glossary

Table 2 - Glossary

Term	Description
AAA	Authentication, Authorization, and Accounting: • Authentication - verifies user credentials (username and password) • Authorization - grants or refuses privileges to a user/client for accessing specific services • Accounting - tracks network resources consumption by users
ARP	Address Resolution Protocol. A protocol that translates IP addresses into MAC addresses for communication over a local area network (LAN).
CLI	Command Line Interface. A user interface in which you type commands at the prompt.
DCB	Data Center Bridging.
DCBX	DCBX protocol is an extension of the Link Layer Discovery Protocol (LLDP). DCBX end points exchange request and acknowledgment messages. For flexibility, parameters are coded in a type-length-value (TLV) format.
DHCP	The Dynamic Host Configuration Protocol (DHCP) is an automatic configuration protocol used on IP networks.

Table 2 - Glossary

Term	Description
DNS	Domain Name System. A hierarchical naming system for devices in a computer network.
ETS	ETS provides a common management framework for assignment of bandwidth to traffic classes.
Gateway	A network node that interfaces with another network using a different network protocol.
HA (High Availability)	A system design protocol that provides redundancy of system components, thus enables overcoming single or multiple failures in minimal downtime.
IB	InfiniBand.
LACP	Link Aggregation Control Protocol (LACP) provides a method to control the bundling of several physical ports together to form a single logical channel. LACP allows a network device to negotiate an automatic bundling of links by sending LACP packets to the peer (directly connected device that also implements LACP).
LDAP	The Lightweight Directory Access Protocol is an application protocol for reading and editing directories over an IP network.
LID (Local IDentifier)	A 16 bit address assigned to end nodes by the subnet manager Each LID is unique within its subnet.
LLDP	Link Layer Discovery Protocol (LLDP) is an industry standard protocol designed to supplant proprietary Link-Layer protocols such as Extreme's EDP (Extreme Discovery Protocol) and CDP (Cisco Discovery Protocol). The goal of LLDP is to provide an inter-vendor compatible mechanism to deliver Link-Layer notifications to adjacent network devices.
MAC	A Media Access Control address (MAC address) is a unique identifier assigned to network interfaces for communications on the physical network segment. MAC addresses are used for numerous network technologies and most IEEE 802 network technologies including Ethernet.
MTU	The maximum size of a packet payload (not including headers) that can be sent /received from a port.
PFC/FC	Priority Based Flow Control applies pause functionality to traffic classes OR classes of service on the Ethernet link.
QoS or Quality of Service	Quality of service is the ability to manage different applications or users by priority such that a required bit rate, delay, packet dropping probability, and/or other measures may be guaranteed.
RADIUS	Remote Authentication Dial In User Service. A networking protocol that enables AAA centralized management for computers to connect and use a network service.

Table 2 - Glossary

Term	Description
RDMA	Remote Direct Memory Access. Accessing memory in a remote side without involvement of the remote CPU.
RSTP	Rapid Spanning Tree Protocol. A spanning-tree protocol used to prevent loops in bridge configurations. RSTP is not aware of VLANs and blocks ports at the physical level.
SCP	Secure Copy or SCP is a means of securely transferring computer files between a local and a remote host or between two remote hosts. It is based on the Secure Shell (SSH) protocol.
SM	Subnet Manager. An entity that configures and manages the subnet, discovers the network topology, assign LIDs, determines the routing schemes and sets the routing tables. There is only one master SM and possible several slaves (Standby mode) at a given time. The SM administers switch routing tables thereby establishing paths through the fabric.
SNMP	Simple Network Management Protocol. A network protocol for the management of a network and the monitoring of network devices and their functions.
SNTP	Network Time Protocol. A protocol for synchronizing computer clocks in a network.
SSH	Secure Shell. A protocol (program) for securely logging in to and running programs on remote machines across a network. The program authenticates access to the remote machine and encrypts the transferred information through the connection.
syslog	A standard for forwarding log messages in an IP network.
TACACS+	Terminal Access Controller Access-Control System Plus. A networking protocol that enables access to a network of devices via one or more centralized servers. TACACS+ provides separate AAA services.
TCA (Target Channel Adapter)	A Channel Adapter that is not required to support verbs, usually used in I/O devices.
VSR	Virtual switch router
WebUI	Web User Interface. A user interface in which you select commands from drop down menus or by clicking on icons.
XML Gateway	Extensible Markup Language Gateway. Provides an XML request-response protocol for setting and retrieving information.

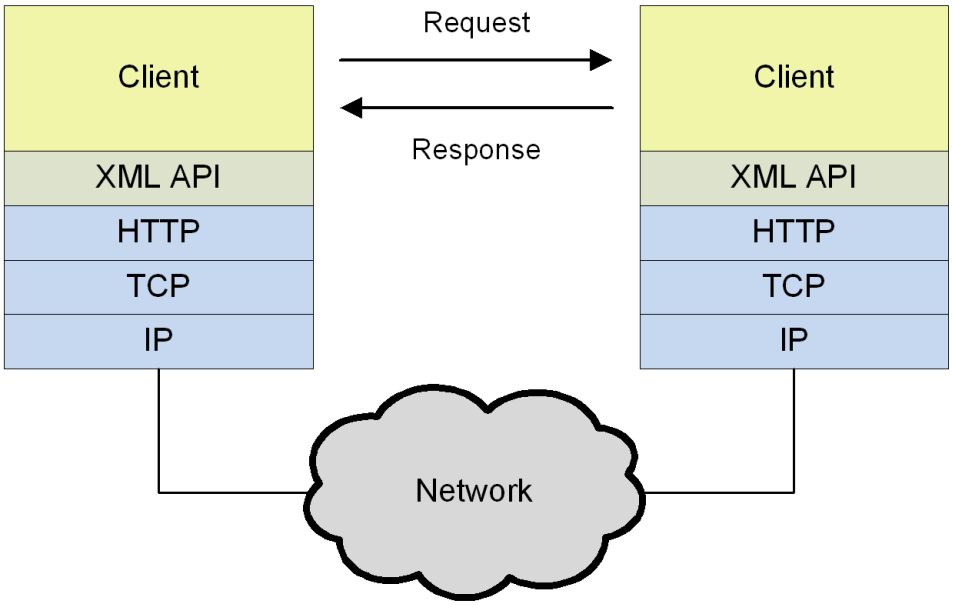
1 Introduction

The MLNX-OS® XML API consists of a virtual tree of nodes. Each node represents a different parameter or table that could be configured or retrieved via the API.

The “XML API” user interface has been developed in order to allow the user to preform operations and query data in the system in a REST-like object oriented API. This UI allows, for example, performing operations such as deleting or changing attributes on the available supported system objects. These objects are called XML nodes and are identified by their node path.

This chapter describes the possible transport layers to use the XML API. In addition, it defines the format and supplies basic examples. The rest of this book defines a list of nodes in the XML tree to be used by the XML API client.

Figure 1: Layered Architecture of XML API



1.1 Transport Layer

1.1.1 XML Over HTTP/HTTPS

The XML API protocol can run over HTTP/HTTPS and use the existing web authentication mechanism.

In order to access the system via HTTP/HTTPS the user needs to use an HTTP/HTTPS client to send POST requests to the system.

1.1.1.1 Authentication

The HTTP client must first authenticate by sending a POST request to the following URL:

```
http://<IP Adress>/admin/launch?script=rh&template=login&action=login
```

The POST request content should contain the following variables:

- f_user_id
- f_password

See usage example below in section "HTTP/HTTPS usage example"

1.1.1.2 Sending the Request

The user needs to construct a POST request containing XML request they wish to execute as the payload of the request. Using an HTTP client the POST request should be sent to the following URL:

```
http://<IP Address>/xtree
```

After the request will be handled in the system the HTTP client will receive a reply with indication of the request execution result.

See section "Request Format" for the request format

See section "Response Format" for the response format

See usage example below in section "HTTP/HTTPS usage example"

1.1.1.3 HTTP/HTTPS Usage Example

The following example use the cURL tool (common in Linux systems) to send XML http POST requests to the system.

Step 1. Create the XML API request in a file. For example see "example.xml" file below:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
  <xg-request>
    <action-request>
        action-name>/do_rest</action-name>
        <nodes>
        <node>
        <name>get</name>
        <type>string</type>
        <value>/mlnxos/api_version</value>
        </node>
     </nodes>
  </action-request>
</xg-request>
```

Step 2. Authentication and establishment of a session Id

```
curl -c /tmp/cookie -d "f_user_id=xmladmin&f_password=xmladmin" http://10.10.10.10/
admin/launch?script=rh&template=login&action=login
```

Step 3. Send the POST request
Run the following from your server shell to execute the request in the file example.xml in the system

```
curl -b /tmp/cookie -X POST -d @example.xml "http://10.10.10.10/xtree"
```

1.1.2 XML Over SSH

The XML API can be used over SSH protocol (instead of HTTP), it then uses the existing CLI authentication mechanism. In order to access the system via SSH the user needs to use an SSH client to establish the login to the system.

1.1.2.1 Authentication

Authentication is done using standard SSH client. Use only the "xmladmin" username when in such case.

When prompt for a password enter the password for the "xmladmin" username (default is "xml-admin")

See usage example below in section "SSH usage example".

1.1.2.2 Sending the request

The user needs to direct the XML request (from a file) to the SSH session established.

After the request will be handled in the system a reply with indication of the request execution result will be given.

See section "Request Format" for the request format

See section "Response Format" for the response format

See usage example below in section "SSH usage example"

1.1.2.3 SSH usage example

The following example uses the pipeline operator ("|") to direct the XML request to the established SSH session

Step 1. Create the XML API request in a file. For example see "example.xml" file at section XML request - "example.xml"

Step 2. Send the file via the SSH connection, run the following from your server shell to execute the request example.xml in the system

```
cat example.xml | ssh xmladmin@<switch_ip>
```

The execution result will be similar to this:

```
# cat example.xml | ssh xmladmin@10.10.10.10
Pseudo-terminal will not be allocated because stdin is not a terminal.
Warning: Permanently added '10.10.10.10' (RSA) to the list of known hosts.

Mellanox MLNX-OS Switch Management
<?xml version="1.0" encoding="UTF-8"?>
<xg-response>
<action-response>
<return-status>
<return-code>0</return-code>
<return-msg></return-msg>
</return-status>
<nodes>
<node>
<name>/mlnxos/api_version</name>
<type>string</type>
<value>1.16</value>
</node>
</nodes>
</action-response>
</xg-response>
```

Use only the xmladmin username in such case (password is “xmladmin”).

1.2 API Format

The XML API request can be sent by an HTTP or HTTPS post request or by using SSH. In either case, all the nodes and values for the various operations are encoded in bindings. Multiple requests may be sent via one XML API request operation.

In order to use the delimiter “/” (forward-slash) in the value, it should be replaced by “\/” (backslash, forward-slash)

1.2.1 Request Format

The request format consists of a list of nodes. Each node represents a possible operation that can be performed on the switch. The XML API request format structure is as follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
  <xg-request>
   <action-request>
    <action-name>/do_rest</action-name>
    <nodes>
<!-- Action node -->
     <node>
      <name>action</name>
      <type>string</type>
      <value>[action_path]|[arg_name]=[arg_value]|[arg_name]=[arg_value]|[…]</value>
     </node>
<!-- Get node -->
     <node>
      <name>get</name>
      <type>string</type>
      <value>[get_path]</value>
     </node>
<!-- Set-create node -->
     <node>
      <name>set-create</name>
      <type>string</type>
      <value>[set_create_path]</value>
     </node>
<!-- Set-delete node -->
     <node>
      <name>set-delete</name>
      <type>string</type>
      <value>[set_delete_path]</value>
     </node>
<!-- Set-modify node -->
     <node>
      <name>set-modify</name>
      <type>string</type>
      <value>[set_modify_path]=[set_value]</value>
     </node>
    </nodes>
   </action-request>
 </xg-request>
```

1.2.2 Response Format

The HTTP POST response format structure is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
  <xg-response>
   <action-response>
    <return-status>
     <return-code>[error_code]</return-code>
     <return-msg>line:[line_number]|[error_msg]</return-msg>
    </return-status>
    <nodes>
     <node>
      <name>[node_path]</name>
      <type>[node_type]</type>
      <value>[node_value]</value>
     </node>
     <node>
      <name>[node_path]</name>
      <type>[node_type]</type>
      <value>[node_value]</value>
     </node>
     <node>
      <name>[node_path]</name>
      <type>[node_type]</type>
      <value>[node_value]</value>
     </node>
      :
      :
      :
    </nodes>
   </action-response>
  </xg-response>
```

1.2.3 Operations and Arguments

MLNX-OS exposes virtual nodes which are served over an ‘action’ operation. It implements an internal logic, which is content-aware in order to enable more flexibility in defining the API in an object-oriented manner.

The action-request name is fixed to ‘/do_rest’.

Action request nodes can have the operation modes (via the ‘name’ parameter) specified in the subsections below.

A full XML request example using all the nodes is also provided in Section 1.2.5, “Example,” on page 18.

1.2.3.1 Get Operations

get – retrieves information from the system. You may query a specific node or iterate and query a node hierarchy using the star character (“*”). A single star queries the first level hierarchy, and two stars (“**”) queries the child nodes of that hierarchy as well.

Get operations can receive the following arguments:

- name = get
- type = string
- value = “[get_path]”

Examples:

/mlnxos/api_version
/mlnxos/v1/chassis/snmp/trap_destinations/*
/mlnxos/v1/chassis/fans/**

1.2.3.2 Set-Create Operations

Set-create creates a new configuration on the system. This operation adds new values to configurable wild card nodes.

Arguments for performing a ‘set-create’ operation:

- name = set-create
- type = string
- value = “[set_create_path]”

Example:

/mlnxos/v1/chassis/snmp/trap_destinations/10.10.10.10

1.2.3.3 Set-Delete Operations

Set-delete deletes an exiting configuration on the system. This operation deletes values of configurable wild card nodes.

Arguments for performing a ‘set-delete’ operation:

- name = set-delete
- type = string
- value = “[set_delete_path]”

Example:

/mlnxos/v1/chassis/snmp/trap_destinations/10.10.10.10

1.2.3.4 Set-Modify Operations

Set-modify changes an exiting configuration on the system. This operation changes values to configurable nodes.

Arguments for performing a ‘set-modify’ operation:

- name = set-modify
- type = string
- value = “[set_modify_path]=[set_value]”

Example:
/mlnxos/v1/chassis/snmp/trap_destinations/10.10.10.10/enable=false

1.2.3.5 Action Operations

Actions execute operations on the system that require parameters.

An action can receive the following arguments:

- name = action
- type = string
- value = “[action_path]|[arg_name]=[arg_value]|[arg_name]=[arg_value]|...”

Example:
/mlnxos/v1/vsr/vsr-default/vlans/add|vlan_id=4

1.2.4 Return Codes

Each XML API request is answered with an XML response. The response contains the following parts:

- return-code – if the request is successful, the return code should be 0. Any other value signifies a failed request.
- return-msg – if the request is successful the return message is empty. If the request fails, a description of the error is given.

  In case of error, the error information is encoded into the return message:

  return-msg = “line:[line_number]|[error_msg]”

In case of a get request, the response also contains the relevant requested nodes. Each node is represented as follows:

- name = [node_path]
- type = [node_type]
- value = [node_value]

1.2.5 Example

In the example below you can see that:

- Node 1 – gets the current API version
- Node 2 – gets all system fans status and speed
- Node 3 – creates a new hosts for listening to SNMP trap
- Node 4 – gets all the hosts listening to SNMP traps
- Node 5 – changes an attribute for one of the hosts listening to SNMP traps
- Node 6 – deletes one of the hosts listening to SNMP traps
- Node 7 – creates a VLAN in the system

Request example:

```
<?xml version="1.0" encoding="utf8"?>
<xg-request>
                  <action-request>
                                  <action-name>/do_rest</action-name>
                                  <nodes>

                                                        <!-- node 1 example - using get oper-
ation -->
                                                        <node>
                                                                        <name>get</name>
                                                                        <type>string</type>
                                                                        <value>/mlnxos/
api_version</value>
                                                        </node>

                                                        <!-- node 2 example - using get oper-
ation with wiled cards  all nodes and their child nodes in the node hierarchy -->
                                                        <node>
                                                                        <name>get</name>
                                                                        <type>string</type>
                                                                        <value>/mlnxos/v1/
chassis/fans/**</value>
                                                        </node>

                                                        <!-- node 3 example - using set-cre-
ate operation -->
                                                        <node>
                                                                        <name>set-create</
name>
                                                                        <type>string</type>
                                                                        <value>/mlnxos/v1/
chassis/snmp/trap_destinations/10.10.10.10</value>
                                                        </node>

                                                        <node>
                                                                        <name>set-create</
name>
                                                                        <type>string</type>
                                                                        <value>/mlnxos/v1/
chassis/snmp/trap_destinations/20.20.20.20</value>
                                                        </node>
```

```
                                                <!-- node 4 example - using get oper-
ation with wiled cards - all nodes in first level of hierarchy -->
                                                <node>
                                                        <name>get</name>
                                                        <type>string</type>
                                                        <value>/mlnxos/v1/
chassis/snmp/trap_destinations/*</value>
                                                </node>

                                                <!-- node 5 example - using set-mod-
ify operation -->
                                                <node>
                                                        <name>set-modify</
name>
                                                        <type>string</type>
                                                        <value>/mlnxos/v1/
chassis/snmp/trap_destinations/10.10.10.10/enable=false</value>
                                                </node>

                                                <!-- node 6 example - using set-
delete operation -->
                                                <node>
                                                        <name>set-delete</
name>
                                                        <type>string</type>
                                                        <value>/mlnxos/v1/
chassis/snmp/trap_destinations/10.10.10.10</value>
                                                </node>

                                                <!-- node 7 example - using action
operation -->
                                                <node>
                                                        <name>action</name>
                                                        <type>string</type>
                                                        <value>/mlnxos/v1/
vsr/vsr-default/vlans/add|vlan_id=4</value>
                                                </node>

                                        </nodes>
                        </action-request>
</xg-request>
```

Response example:

```
<?xml version="1.0" encoding="UTF-8"?>
<xg-response>
  <action-response>
    <return-status>
      <return-code>0</return-code>
      <return-msg></return-msg>
    </return-status>
    <nodes>
      <node>
        <name>/mlnxos/api_version</name>
        <type>string</type>
        <value>1.16</value>
      </node>
      <node>
        <name>/mlnxos/v1/chassis/fans/FAN1\/FAN</name>
        <type>string</type>
        <value>FAN1/FAN</value>
      </node>
      <node>
        <name>/mlnxos/v1/chassis/fans/FAN1\/FAN/1</name>
        <type>uint32</type>
        <value>1</value>
      </node>
      <node>
        <name>/mlnxos/v1/chassis/fans/FAN1\/FAN/1/speed</name>
        <type>float64</type>
        <value>10260</value>
      </node>
      <node>
        <name>/mlnxos/v1/chassis/fans/FAN1\/FAN/1/status</name>
        <type>string</type>
        <value>OK</value>
      </node>
      <node>
        <name>/mlnxos/v1/chassis/fans/FAN1\/FAN/2</name>
        <type>uint32</type>
        <value>2</value>
      </node>
      <node>
        <name>/mlnxos/v1/chassis/fans/FAN1\/FAN/2/speed</name>
        <type>float64</type>
        <value>10260</value>
      </node>
      <node>
        <name>/mlnxos/v1/chassis/fans/FAN1\/FAN/2/status</name>
        <type>string</type>
        <value>OK</value>
      </node>
```

```
<node>
  <name>/mlnxos/v1/chassis/fans/FAN2\/FAN</name>
  <type>string</type>
  <value>FAN2/FAN</value>
</node>
<node>
  <name>/mlnxos/v1/chassis/fans/FAN2\/FAN/1</name>
  <type>uint32</type>
  <value>1</value>
</node>
<node>
  <name>/mlnxos/v1/chassis/fans/FAN2\/FAN/1/speed</name>
  <type>float64</type>
  <value>10560</value>
</node>
<node>
  <name>/mlnxos/v1/chassis/fans/FAN2\/FAN/1/status</name>
  <type>string</type>
  <value>OK</value>
<node>
<node>
  <name>/mlnxos/v1/chassis/fans/FAN2\/FAN/2</name>
  <type>uint32</type>
  <value>2</value>
</node>
<node>
  <name>/mlnxos/v1/chassis/fans/FAN2\/FAN/2/speed</name>
  <type>float64</type>
  <value>10830</value>
</node>
<node>
  <name>/mlnxos/v1/chassis/fans/FAN2\/FAN/2/status</name>
  <type>string</type>
  <value>OK</value>
</node>
<node>
  <name>/mlnxos/v1/chassis/fans/PS1\/FAN</name>
  <type>string</type>
  <value>PS1/FAN</value>
</node>
<node>
  <name>/mlnxos/v1/chassis/fans/PS1\/FAN/1</name>
  <type>uint32</type>
  <value>1</value>
</node>
<node>
  <name>/mlnxos/v1/chassis/fans/PS1\/FAN/1/speed</name>
  <type>float64</type>
  <value>-1</value>
</node>
```

```
      <node>
        <name>/mlnxos/v1/chassis/fans/PS1\/FAN/1/status</name>
        <type>string</type>
        <value>NOT PRESENT</value>
      </node>
      <node>
        <name>/mlnxos/v1/chassis/fans/PS2\/FAN</name>
        <type>string</type>
        <value>PS2/FAN</value>
      </node>
      <node>
        <name>/mlnxos/v1/chassis/fans/PS2\/FAN/1</name>
        <type>uint32</type>
        <value>1</value>
      </node>
      <node>
        <name>/mlnxos/v1/chassis/fans/PS2\/FAN/1/speed</name>
        <type>float64</type>
        <value>11160</value>
      </node>
      <node>
        <name>/mlnxos/v1/chassis/fans/PS2\/FAN/1/status</name>
        <type>string</type>
        <value>OK</value>
      </node>
      <node>
        <name>/mlnxos/v1/chassis/snmp/trap_destinations/10.10.10.10</name>
        <type>hostname</type>
        <value>10.10.10.10</value>
      </node>
      <node>
        <name>/mlnxos/v1/chassis/snmp/trap_destinations/20.20.20.20</name>
        <type>hostname</type>
        <value>20.20.20.20</value>
      </node>
    </nodes>
  </action-response>
</xg-response>
```

1.3 Legacy API Format

The MLNX-OS XML API has a legacy API format that had been in use before the development of the current do_REST API format (described in Section 1.2, “API Format,” on page 14). This section goes over the usage and format of the legacy API.

The request format of the legacy API consists of single operation type. The supported operations are queries and actions.

1.3.1 Legacy Query Request Format

This operation queries information from the system based on the node (or nodes) given in the query request.

1.3.1.0.1 Query Request via HTTP POST or SSH Transport Layer

A POST or SSH query request may be for a single node or multiple nodes. This operation queries information from the system based on the node given in the “name” tag of the requested node. A query request can also add the tags “subop” and “iterate” to iterate the values of a node.

When adding the “iterate” tag, the user can also add a flag to specify the depth of the iteration:

- If no flags are used, a shallow iteration is performed
- If the “subtree” flag is used, the subtree is queried
- If the “include-self” flag is used, the subtree is queried including the node itself

Query request structure format is as follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<xg-request>
    <query-request>
            <nodes>
                  <node>
                        <name>[full node path]</name>
                        <!-- to iterate a node add subop -->
                        <subop>iterate</subop>
                        <!-- to control the depth of the iteration add the flags tag -
->
                        <flags>
                         <flag>subtree</flag>
                         <flag>include-self</flag>
                        </flags>

                  </node>
                  ...
                  <node>
                  </node>
            </nodes>
    </query-request>
</xg-request>
```

1.3.1.0.2 Query Request via HTTP GET Transport Layer

*****IMPORTANT NOTE*****

All information retrieving can be done using the legacy API format. In other words, reading ro nodes data, reading rw nodes data or reading rc nodes data can be done using both the legacy API format and the regular API format (do_rest format).

The legacy API also supports passing query request as an http get request. The HTTP GET interface supports only a single node/node root.

With HTTP get request the user can have control over the iteration of nodes as follows:

- If “/*” is at the end of a node and is appended the flag: “include –self”, a shallow iteration is performed (see example below)
- If “/**” is at the end of a node, the subtree is queried
- If “/***” is at the end of a node, the subtree is queried including the node itself

HTTP GET request are sent to a URL as shown in the following:

```
http://<hostname>/xtree/<node-path>
```

For example, this request queries the XML API version number:

```
http://10.10.10.10/xtree/mlnxos/api_version
```

This request queries all the system fans and their attributes:

```
http://10.10.10.10/xtree/mlnxos/v1/chassis/fans/***
```

This request queries all the specific chassis attributes:

```
http://10.10.10.10/xtree/mlnxos/v1/chassis/*&flag=include-self
```

If HTTP GET requests are viewed in a web browser, they are automatically be styled using XSLT and the resulting links become clickable for somewhat limited navigation.

1.3.2 Legacy Action Request Format

*****IMPORTANT NOTE*****

The action nodes listed in Section 2.7, “Legacy API Nodes,” on page 179 can be used only in the legacy API format.

Actions call the requested node with the given parameters and execute the relevant operation on the system.

Action request are supported only via HTTP post or SSH transport layers.

Action request format structure is as follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<xg-request>
    <action-request>

        <!-- action node name -->
        <action-name>[full node path]</action-name>

        <!-- action parameters listed as nodes-->
                        <nodes>
          <node>
             <name> [name of argument] </name>
             <type> [type of argument] </type>
             <value> [value of argument]</value>
          </node>
        </nodes>

    </action-request>
</xg-request>
```

1.3.3 Response Format and Return Codes

The response and return codes for a legacy API request is the same as the regular API response (see Section 1.3.4, “Legacy Request Examples”)

1.3.4 Legacy Request Examples

In the example below are two requests:

- The first request is an action request to download an image file to the system
- The second request is a query request to see the current state of the download (“running”, “completed”, etc). In addition, the request is iterated for the system fans and their attributes.

Action request example:

```
<?xml version="1.0" encoding="utf8"?>

<xg-request>
    <action-request>

        <action-name>/mlnxos/v1/chassis/file_transfer/download</action-name>
                                    <nodes>
                                                    <node>
                                                                    <name> remote_url </
name>
                                                                    <type> uri </type>
                                                                    <value>scp://
user:password@hostname/file_path/image_file.img</value>
                                                    </node>
                                                    <node>
                                                                    <name> local_file-
name </name>
                                                                    <type> string </
type>
                                                                    <value>my_-
file_name.img</value>
                                                    </node>
                                                    <node>
                                                                    <name> type </name>
                                                                    <type> string </
type>
                                                                    <value>img</value>
                                                    </node>
                                    </nodes>

    </action-request>
</xg-request>
```

Action response example:

```
<?xml version="1.0" encoding="UTF-8"?>
<xg-response>
  <action-response>
    <return-status>
      <return-code>0</return-code>
      <return-msg></return-msg>
    </return-status>
  </action-response>
</xg-response>
```

Query request example:

```
<?xml version="1.0" encoding="utf8"?>

<xg-request>
    <query-request>

        <nodes>
            <node>
                <name>/mlnxos/v1/chassis/file_transfer/state/download/state</name>
            </node>

            <node>
                <name>/mlnxos/v1/chassis/fans</name>
                <subop>iterate</subop>
                <flags>
                    <flag>subtree</flag>
                    <flag>include-self</flag>
                </flags>
            </node>
        </nodes>

    </query-request>
</xg-request>
```

Query response example:

```
<?xml version="1.0" encoding="UTF-8"?>
<xg-response>
  <query-response>
    <return-status>
      <return-code>0</return-code>
      <return-msg></return-msg>
    </return-status>
    <db-revision-id>99</db-revision-id>
    <nodes>
      <node>
        <name>/mlnxos/v1/chassis/fans/FAN1\/FAN</name>
        <type>string</type>
        <value>FAN1/FAN</value>
      </node>
      <node>
        <name>/mlnxos/v1/chassis/fans/FAN1\/FAN/1</name>
        <type>uint32</type>
        <value>1</value>
      </node>
      <node>
        <name>/mlnxos/v1/chassis/fans/FAN1\/FAN/1/speed</name>
        <type>float64</type>
        <value>10260</value>
      </node>
      <node>
        <name>/mlnxos/v1/chassis/fans/FAN1\/FAN/1/status</name>
        <type>string</type>
        <value>OK</value>
      </node>
      <node>
        <name>/mlnxos/v1/chassis/fans/FAN1\/FAN/2</name>
        <type>uint32</type>
        <value>2</value>
      </node>
      <node>
        <name>/mlnxos/v1/chassis/fans/FAN1\/FAN/2/speed</name>
        <type>float64</type>
        <value>10560</value>
      </node>
      <node>
        <name>/mlnxos/v1/chassis/fans/FAN1\/FAN/2/status</name>
        <type>string</type>
        <value>OK</value>
      </node>
      <node>
        <name>/mlnxos/v1/chassis/fans/FAN2\/FAN</name>
        <type>string</type>
        <value>FAN2/FAN</value>
      </node>
```

```
<node>
  <name>/mlnxos/v1/chassis/fans/FAN2\/FAN/1</name>
  <type>uint32</type>
  <value>1</value>
</node>
<node>
  <name>/mlnxos/v1/chassis/fans/FAN2\/FAN/1/speed</name>
  <type>float64</type>
  <value>10830</value>
</node>
<node>
  <name>/mlnxos/v1/chassis/fans/FAN2\/FAN/1/status</name>
  <type>string</type>
  <value>OK</value>
</node>
<node>
  <name>/mlnxos/v1/chassis/fans/FAN2\/FAN/2</name>
  <type>uint32</type>
  <value>2</value>
</node>
<node>
  <name>/mlnxos/v1/chassis/fans/FAN2\/FAN/2/speed</name>
  <type>float64</type>
  <value>10560</value>
</node>
<node>
  <name>/mlnxos/v1/chassis/fans/FAN2\/FAN/2/status</name>
  <type>string</type>
  <value>OK</value>
</node>
<node>
  <name>/mlnxos/v1/chassis/fans/PS1\/FAN</name>
  <type>string</type>
  <value>PS1/FAN</value>
</node>
<node>
  <name>/mlnxos/v1/chassis/fans/PS1\/FAN/1</name>
  <type>uint32</type>
  <value>1</value>
</node>
<node>
  <name>/mlnxos/v1/chassis/fans/PS1\/FAN/1/speed</name>
  <type>float64</type>
  <value>-1</value>
</node>
<node>
  <name>/mlnxos/v1/chassis/fans/PS1\/FAN/1/status</name>
  <type>string</type>
  <value>NOT PRESENT</value>
</node>
```

```
      <node>
        <name>/mlnxos/v1/chassis/fans/PS2\/FAN</name>
        <type>string</type>
        <value>PS2/FAN</value>
      </node>
      <node>
        <name>/mlnxos/v1/chassis/fans/PS2\/FAN/1</name>
        <type>uint32</type>
        <value>1</value>
      </node>
      <node>
        <name>/mlnxos/v1/chassis/fans/PS2\/FAN/1/speed</name>
        <type>float64</type>
        <value>11160</value>
      </node>
      <node>
        <name>/mlnxos/v1/chassis/fans/PS2\/FAN/1/status</name>
        <type>string</type>
        <value>OK</value>
      </node>
      <node>
        <name>/mlnxos/v1/chassis/file_transfer/state/download/state</name>
        <type>string</type>
        <value>completed</value>
      </node>
    </nodes>
  </query-response>
</xg-response>
```

2 Supported Nodes

This chapter presents the different XML nodes and specifying their function and usage.

2.1 Node Description

The following fields are detailed for each node:

- Access – the type of operations possible on this node
  - ro – read only nodes to query information from the system
  - rw – read/write nodes to modify configurations and query configurations
  - rc – read/create nodes to create, delete, and query configurations
  - action – to preform operations in the system and cannot be used to query information
- Data type – the data type of this node (e.g. string)
- CLI example – reference to an equivalent CLI command if relevant
- Release version – XML API version in which this node has been added

2.2 Node Types

The following node types are available:

- General nodes – addressed via the prefix /mlnxos, they currently only contain the /mlnxos/api_version node
- Chassis nodes – addressed via the prefix /mlnxos/v1/chassis, they are used for general system configuration and data query
- VSR nodes – VSR represents an instance of a managed switch-router. Currently only a single instance is supported named vsr-default. The VSR nodes are addressed via the prefix /mlnxos/v1/vsr/vsr-default.
- Legacy API nodes – supported in the Legacy API format only (see Section 1.3, “Legacy API Format,” on page 23 for more information)

2.3 General

/mlnxos/api_version

Description	Version of API exposed by device Format: major.minor Example: 1.0
Access	ro
Data Type	string
Release Version	1.0
CLI Example	N/A
Notes	

2.4 Chassis

2.4.1 General

/mlnxos/v1/chassis/model

Description	Model of chassis Example: sx1016
Access	ro
Data Type	string
Release Version	1.0
CLI Example	show inventory
Notes	

/mlnxos/v1/chassis/pn

Description	Chassis part number Example: MIS5200Q-7DNC0000000
Access	ro
Data Type	string
Release Version	1.0
CLI Example	show inventory
Notes	

/mlnxos/v1/chassis/sn

Description	Chassis serial number Example: MT018X001590000000000000
Access	ro
Data Type	string
Release Version	1.0
CLI Example	show inventory
Notes	

/mlnxos/v1/chassis/hostname

Description	Chassis host name Example: switch-11a14e
Access	ro
Data Type	string
Release Version	1.0
CLI Example	show hosts
Notes	

/mlnxos/v1/chassis/sw_version

Description	Chassis SW version Example: X86_64 3.4.2008 2015-06-12 11:48:53 x86_64
Access	ro
Data Type	string
Release Version	1.0
CLI Example	show version [concise]
Notes	

/mlnxos/v1/chassis/next_boot_sw_version

Description	Chassis next boot SW version (The version which is expected to load on next boot) Example: X86_64 3.4.2008 2015-06-12 11:48:53 x86_64
Access	ro
Data Type	string
Release Version	1.0
CLI Example	show images
Notes	

/mlnxos/v1/chassis/uptime

Description	Device uptime (Time passed since boot in msec) Example: 226777824
Access	ro
Data Type	duration_ms
Release Version	1.0
CLI Example	show version
Notes	

/mlnxos/v1/chassis/system_profile

Description	Configure the system profile Optional system profiles (depending on license and system type):ib-single-switch • eth-single-switch • ib-no-adaptive-routing-single-switch • vpi-single-switch
Access	rw
Data Type	string
Release Version	1.17
CLI Example	system profile <profile name> show system profile
Notes	• Changing profile in the system causes a reboot to the switch • Changing profile using this XML node is supported on 1U systems and single management director systems

/mlnxos/v1/chassis/ib/num_of_swids

Description	Queries the number of SWIDs currently available in the system
Access	ro
Data Type	uint8
Release Version	1.19
CLI Example	show system profile
Notes	

/mlnxos/v1/chassis/ib/adaptive_routing

Description	Queries if adaptive routing is enabled in the system
Access	ro
Data Type	bool
Release Version	1.19
CLI Example	show system profile
Notes	

/mlnxos/v1/chassis/ib/ib_router

Description	Queries if IB routing capability is enabled on the system
Access	ro
Data Type	bool
Release Version	1.19
CLI Example	show system profile
Notes	

/mlnxos/v1/chassis/set_system_profile

Description

Changes the system profile.
One of the following profiles may be configured:

- "ib-single-switch"
- "ib-no-adaptive-routing-single-switch"
- "eth-single-switch"
- "vpi-single-switch"
- "ib" profile with additional attributes (as explained in the arguments section)

The available system profiles depend on license and system type.

Arguments

Name	Description	Data Type	Status
profile	The name of the profile to configure	string	mandatory
adaptive_routing	In case of IB profile – enables/disables adaptive routing	bool	optional
ib_routing	In case of IB profile – enables/disables IB router capabilities	bool	optional
num_of_swids	For IB profile – number of SWIDs in the system	unit8	optional

Release Version

1.19

CLI Example

system profile <profile name>
system profile ib <adaptive-routing | ib-router| |num-of-swids>
show system profile

Notes

Changing profile in the system will cause a reboot to the switch
Changing profile using this XML node is supported on 1U systems and single management director systems only

2.4.2 Clustering

/mlnxos/v1/chassis/cluster/local/addr

Description	Management IP address of local device
Access	ro
Data Type	ipv4addr
Release Version	1.0
CLI Example	N/A
Notes	

/mlnxos/v1/chassis/cluster/master/addr

Description	Management IP address of master device
Access	ro
Data Type	ipv4addr
Release Version	1.0
CLI Example	N/A
Notes	

/mlnxos/v1/chassis/cluster/local/role

Description	Role of local device Format: "master", "standby"
Access	ro
Data Type	string
Release Version	1.0
CLI Example	N/A
Notes	

2.4.3 Configuration Database

/mlnxos/v1/chassis/db/reset

Description	Set configuration to factory settings			
Access	action			
Release Version	1.16			
Arguments	Name	Description	Data Type	Status
	import_id	Import ID	string	optional
	backup_d-b_name	Backup filename	string	optional
CLI Example	reset factory [keep-all-config | keep-basic | only-config] [halt]			
Notes				

2.4.4 Fans

/mlnxos/v1/chassis/fans/*

Description	List of fans, main index - name of fan Examples of index: “FAN1\/FAN”, “PS1\/FAN”
Access	ro
Data Type	string
Release Version	1.0
CLI Example	show fan
Notes	Fan names contain a forward-slash (“/”) that is not part of the node hierarchy and, therefore, it is escaped with a backslash (“\”) before the fan name.

/mlnxos/v1/chassis/fans/<STRING>/*

Description	List of fans, secondary index – number of fan
Access	ro
Data Type	uint32
Release Version	1.0
CLI Example	show fan
Notes	

/mlnxos/v1/chassis/fans/<STRING>/<UINT32>/speed

Description	Fan speed in RPM Example: 5400
Access	ro
Data Type	float32
Release Version	1.0
CLI Example	show fan
Notes	

/mlnxos/v1/chassis/fans/<STRING>/<UINT32>/status

Description	Fan status Examples: "NOT PRESENT", "OK"
Access	ro
Data Type	string
Release Version	1.0
CLI Example	show fan
Notes	

2.4.5 Image Management

/mlnxos/v1/chassis/image/boot_location

Description	Query the partition number from which the system booted (last boot partition)
Access	ro
Data Type	uint8
Release Version	1.17
CLI Example	show images
Notes	

/mlnxos/v1/chassis/image/next_boot_location

Description	Query the partition number from which the system will boot (next boot partition)
Access	ro
Data Type	uint8
Release Version	1.17
CLI Example	show images
Notes	

/mlnxos/v1/chassis/image/location/*

Description	List of the partition indexes
Access	ro
Data Type	uint8
Release Version	1.17
CLI Example	show images
Notes	

/mlnxos/v1/chassis/image/location/<UINT8>/version

Description	Query the version installed on the specified partition
Access	ro
Data Type	string
Release Version	1.17
CLI Example	show images
Notes	

2.4.6 Management Interfaces

/mlnxos/v1/chassis/mgmt_interfaces/*

Description	Chassis mgmt. interfaces list
Access	ro
Data Type	ifname
Release Version	1.13
CLI Example	show interfaces <if name>
Notes	

/mlnxos/v1/chassis/mgmt_interfaces/<IFNAME>/enabled

Description	Mgmt. interface administrative state
Access	ro
Data Type	bool
Release Version	1.13
CLI Example	show interface <if-name>
Notes	

/mlnxos/v1/chassis/mgmt_interfaces/<IFNAME>/ipv4/dhcp

Description	DHCP state
Access	ro
Data Type	bool
Release Version	1.13
CLI Example	show interfaces <if name>
Notes	

/mlnxos/v1/chassis/mgmt_interfaces/<IFNAME>/ipv4/addresses/*

Description	List of ipv4 addresses for interface
Access	ro
Data Type	uint8
Release Version	1.13
CLI Example	show interfaces <ifname>
Notes	

/mlnxos/v1/chassis/mgmt_interfaces/<IFNAME>/ipv4/addresses/<UINT8>/address

Description	IPv4 address Example: 10.0.0.1
Access	ro
Data Type	ipv4addr
Release Version	1.13
CLI Example	show interfaces <ifname>
Notes	

/mlnxos/v1/chassis/mgmt_interfaces/<IFNAME>/ipv4/addresses/<UINT8>/mask_len

Description	Mask length
Access	ro
Data Type	uint8
Release Version	1.13
CLI Example	show interfaces <if name>
Notes	

/mlnxos/v1/chassis/mgmt_interfaces/<IFNAME>/ipv6/dhcp

Description	DHCP state
Access	ro
Data Type	bool
Release Version	1.13
CLI Example	show interfaces <if name>
Notes	

/mlnxos/v1/chassis/mgmt_interfaces/<IFNAME>/ipv6/addresses/*

Description	List of ipv6 addresses for interface
Access	ro
Data Type	uint8
Release Version	1.13
CLI Example	show interfaces <if name>
Notes	

/mlnxos/v1/chassis/mgmt_interfaces/<IFNAME>/ipv6/addresses/<UINT8>/address

Description	IPv6 address Example: fe80::8038:2182:6e95:3cd8
Access	ro
Data Type	ipv6addr
Release Version	1.13
CLI Example	show interfaces <if name>
Notes	

/mlnxos/v1/chassis/mgmt_interfaces/<IFNAME>/ipv6/addresses/<UINT8>/mask_len

Description	Mask length
Access	ro
Data Type	uint8
Release Version	1.13
CLI Example	show interfaces <if name>
Notes	

/mlnxos/v1/chassis/mgmt_interfaces/<IFNAME>/physical_address

Description	Query the physical address (MAC address) of a management interface
Access	ro
Data Type	macaddr802
Release Version	1.18
CLI Example	show interfaces <if name>
Notes	

2.4.7 Modules

/mlnxos/v1/chassis/modules/*

Description	List the system modules. For example "MGMT", "PS1", "FAN1".
Access	ro
Data Type	string
Release Version	1.0
CLI Example	show inventory
Notes	

/mlnxos/v1/chassis/modules/<STRING>/part_number

Description	Module part number Examples: "SA000203-B", "MSX60-FF", "MSX6036F-1BFR", "MSX60-PF"
Access	ro
Data Type	string
Release Version	1.0
CLI Example	show inventory
Notes	

/mlnxos/v1/chassis/modules/<STRING>/serial_number

Description	Module serial numberExample: "MT1140X00208"
Access	ro
Data Type	string
Release Version	1.0
CLI Example	show inventory
Notes	

/mlnxos/v1/chassis/modules/<STRING>/status

Description	Module status Examples: "OK", "NOT PRESENT"
Access	ro
Data Type	string
Release Version	1.0
CLI Example	show inventory
Notes	

/mlnxos/v1/chassis/modules/<STRING>/type

Description	Type of module Examples: "CPU", "SXX0XX_FAN", "SX6036", "SXX0XX_PS"
Access	ro
Data Type	string
Release Version	1.0
CLI Example	show inventory
Notes	

2.4.8 Power Supplies

/mlnxos/v1/chassis/power_supplies/*

Description	List of power supplies
Access	ro
Data Type	string
Release Version	1.0
CLI Example	show power
Notes	Relevant for director switches only

/mlnxos/v1/chassis/power_supplies/<PS_Name>/current

Description	Current in ampere
Access	ro
Data Type	float32
Release Version	1.0
CLI Example	show power
Notes	Relevant for director switches only

/mlnxos/v1/chassis/power_supplies/<PS_Name>/power

Description	Power in Watts
Access	ro
Data Type	float32
Release Version	1.0
CLI Example	show power
Notes	Relevant for director switches only

/mlnxos/v1/chassis/power_supplies/<PS_Name>/voltage

Description	Voltage in Volts
Access	ro
Data Type	float32
Release Version	1.0
CLI Example	show voltage
Notes	Relevant for director switches only

2.4.9 SNMP

/mlnxos/v1/chassis/snmp/trap_destinations/*

Description	Trap destinations list for SNMP Example: 10.0.0.1
Access	rc
Data Type	hostname
Release Version	1.16
CLI Example	snmp-server host <host ip>
Notes	

/mlnxos/v1/chassis/snmp/trap_destinations/<IPADDR>/community

Description	Community for traps Example: "public"
Access	rw
Data Type	string
Release Version	1.16
CLI Example	snmp-server host <ip> traps <community name> show snmp host
Notes	

/mlnxos/v1/chassis/snmp/trap_destinations/<IPADDR>/enable

Description	Traps enablement to destination
Access	rw
Data Type	bool
Release Version	1.16
CLI Example	[no] snmp-server host <ip> disable
Notes	Options are: "true" or "false"

/mlnxos/v1/chassis/snmp/trap_destinations/<IPADDR>/port

Description	Destination port number for traps Example: 162
Access	rw
Data Type	uint16
Release Version	1.6
CLI Example	snmp-server host <ip> traps port <port num> show snmp host
Notes	

/mlnxos/v1/chassis/snmp/trap_destinations/<IPADDR>/type

Description	Trap type Format: "trap-v1", "trap-v2c"
Access	rw
Data Type	string
Release Version	1.16
CLI Example	snmp-server host <ip> traps version <version>
Notes	trap types are: • trap-v1 • trap-v2c • inform-v2c • trap-v3 • inform-v3

/mlnxos/v1/chassis/snmp/trap_destinations/<IPADDR>/v3/user

Description	Username of trap sink
Access	rw
Data Type	string
Release Version	1.16
CLI Example	snmp-server host <host-ip> traps version 3 user <username> auth <sha | md5> <password> priv <aes-128|des> <password> snmp-server host <host-ip> informs version 3 engineID <engine-id> user <username> auth <sha|md5> <password> show snmp host
Notes	

/mlnxos/v1/chassis/snmp/trap_destinations/<IPADDR>/v3/inform_engine_id

Description	Engine ID of inform trap sink
Access	rw
Data Type	string
Release Version	1.16
CLI Example	snmp-server host <host-ip> informs version 3 engineID <engine-id> user <username> auth <sha | md5> <password> show snmp host
Notes	

/mlnxos/v1/chassis/snmp/trap_destinations/<IPADDR>/v3/hash_type

Description	Hashing algorithm for authentication password
Access	rw
Data Type	string
Release Version	1.16
CLI Example	snmp-server host <host-ip> traps version 3 user <user name> auth <sha | md5> <password> priv <aes-128|des> <password> snmp-server host <host-ip> informs version 3 engineID <engine-id> user <username> auth <sha|md5> <password> show snmp host
Notes	

/mlnxos/v1/chassis/snmp/trap_destinations/<IPADDR>/v3/auth_key

Description	Encrypted authentication key
Access	rw
Data Type	string
Release Version	1.16
CLI Example	snmp-server host <host ip> traps version 3 user <user name> auth <sha|md5> <password> priv <aes-128|des> <password> snmp-server host <host ip> informs version 3 engineID <engine id> user <user name> auth <sha|md5> <password> show snmp host
Notes	

/mlnxos/v1/chassis/snmp/trap_destinations/<IPADDR>/v3/privacy_type

Description	Hashing algorithm for privacy
Access	rw
Data Type	string
Release Version	1.16
CLI Example	snmp-server host <host-ip> traps version 3 user <user name> auth <sha | md5> <password> priv <aes-128|des> <password> snmp-server host <host-ip> informs version 3 engineID <engine-id> user <username> auth <sha|md5> <password> show snmp host
Notes	

/mlnxos/v1/chassis/snmp/trap_destinations/<IPADDR>/v3/privacy_key

Description	Encrypted privacy key
Access	rw
Data Type	string
Release Version	1.16
CLI Example	snmp-server host <host-ip> traps version 3 user <user name> auth <sha | md5> <password> priv <aes-128|des> <password> snmp-server host <host-ip> informs version 3 engineID <engine-id> user <username> auth <sha|md5> <password> show snmp host
Notes	

2.4.10 Temperature Sensors

/mlnxos/v1/chassis/temperature/*

Description	List of temperature sensors Examples: "/BOARD_MONITOR", "/QSFP_TEMP1", "/SX", "/L05/SX"
Access	ro
Data Type	string
Release Version	1.0
CLI Example	show temperature
Notes	Temperature sensors names can contain a "/" that is not a part of the node hierarchy. Therefore, the "/" needs to be escaped with a backslash ("\") in the node name (e.g.: /mlnxos/v1/chassis/temperature/\/L05\/SX).

/mlnxos/v1/chassis/temperature/<STRING>/temperature

Description	Temperature in Celsius degrees Example: 35
Access	ro
Data Type	float32
Release Version	1.0
CLI Example	show temperature
Notes	

2.5 Ethernet Configuration

2.5.1 Spanning Tree Protocol

/mlnxos/v1/vsr/<STRING>/stp/enabled

Description	Enable/disable Spanning Tree
Access	rw
Data Type	bool
Release Version	1.16
CLI Example	spanning-tree show spanning-tree
Notes	

2.5.2 Routing

/mlnxos/v1/vsr/<STRING>/ip/routing/enabled

Description	Enable/disable IP routing
Access	rw
Data Type	bool
Release Version	1.16
CLI Example	ip routing show ip routing
Notes	L3 must be enabled on the system to operate IP routing features. See the command “ip l3” for more details.

/mlnxos/v1/vsr/<STRING>/ip/ospf/enabled

Description	Enable/disable OSPF
Access	rw
Data Type	bool
Release Version	1.16
CLI Example	protocol ospf show protocols
Notes	L3 must be enabled on the system to operate OSPF features. See the command “ip l3” for more details.

/mlnxos/v1/vsr/<STRING>/ip/ospf/<STRING>/area/<IP STRING>/stub

Description	Configure an area as an OSPF stub area
Access	rw
Data Type	bool
Release Version	1.16
CLI Example	router ospf area <area num or ip> stub show ip ospf
Notes	System currently supports only one instance of OSPF called “default” L3 must be enabled on the system to operate OSPF features. See the command “ip l3” for more details.

2.5.2.1 ACLs

/mlnxos/v1/vsr/<STRING>/acls/ipv4_lists/*

Description	List of IPv4 ACLs
Access	ro
Data Type	string
Release Version	1.16
CLI Example	show ipv4 access-lists summary
Notes	

/mlnxos/v1/vsr/<STRING>/acls/ipv4_lists/add

Description	Add IPv4 ACLs			
Access	action			
Release Version	1.16			
Arguments	Name	Description	Data Type	Status
	acl_name	ACL name	string	mandatory
CLI Example	[no] ipv4 access-list <ipv4-access-list name>			
Notes				

/mlnxos/v1/vsr/<STRING>/acls/ipv4_lists/delete

Description	Delete IPv4 ACLs			
Access	action			
Release Version	1.16			
Arguments	Name	Description	Data Type	Status
	acl_name	ACL name	string	mandatory
CLI Example	[no] ipv4 access-list <ipv4-access-list name>			
Notes				

/mlnxos/v1/vsr/<STRING>/acls/ipv4_lists/<STRING>/rules/*

Description	List of IPv4 ACL rules
Access	ro
Data Type	uint16
Release Version	1.16
CLI Example	show ipv4 access-lists summary
Notes	

/mlnxos/v1/vsr/<STRING>/acls/ipv4_lists/<STRING>/rules/add

Description	Add rule to the specified IPv4 ACL list
Access	action
Release Version	1.16

Arguments

Name	Description	Data Type	Status
rule_type	Rule type Format: "permit", "deny"	string	mandatory
sequence_num	Rule sequence number	uint16	mandatory
protocol	Protocol number Format: "tcp", "udp", "ip"	string	mandatory
acl_action_name	ACL action name (can be "none")	string	mandatory
source_ip	Source IP address (can be "any")	string	mandatory
source_ip_mask	Source IP address mask. If IP is not "any" it is mandatory.	string	mandatory
source_port	Source port number (can be "none")	uint16	mandatory
destination_ip	Destination IP address (can be "any")	string	mandatory
destination_ip_mask	Destination IP address mask. If IP is not "any" it is mandatory.	string	mandatory
destination_port	Destination port number (can be "none")	uint16	mandatory

CLI Example	[seq-number <sequence-number>] {permit | deny} {tcp | udp} {<source-ip> [mask <ip>] | [any]} {<dest-ip> [mask <ip>]| [any]} [eq-source <source_port_number>] [eq-destination <destination_ port_number>] [action <action-id>]
Notes	

/mlnxos/v1/vsr/<STRING>/acls/ipv4_lists/<STRING>/rules/delete

Description	Delete IPv4 ACL rule			
Access	action			
Release Version	1.16			
Arguments	Name	Description	Data Type	Status
	sequence_num	Rule sequence number	uint16	mandatory
CLI Example	ipv4 access-list <list name> no <rule seq num>			
Notes				

/mlnxos/v1/vsr/<STRING>/acls/ipv4_lists/<STRING>/rules/<UINT16>/action_name

Description	ACL action name (can be "none")
Access	ro
Data Type	string
Release Version	1.16
CLI Example	show ipv4 access-lists <list name>
Notes	

/mlnxos/v1/vsr/<STRING>/acls/ipv4_lists/<STRING>/rules/<UINT16>/destination_ip

Description	Destination IP address (can be "any")
Access	ro
Data Type	string
Release Version	1.16
CLI Example	show ipv4 access-lists <list name>
Notes	

/mlnxos/v1/vsr/<STRING>/acls/ipv4_lists/<STRING>/rules/<UINT16>/destination_ip_mask

Description	Destination IP mask (can be "any")
Access	ro
Data Type	string
Release Version	1.16
CLI Example	show ipv4 access-lists <list name>
Notes	

/mlnxos/v1/vsr/<STRING>/acls/ipv4_lists/<STRING>/rules/<UINT16>/destination_port

Description	Destination port number (can be "any")
Access	ro
Data Type	string
Release Version	1.16
CLI Example	show ipv4 access-lists <list name>
Notes	

/mlnxos/v1/vsr/<STRING>/acls/ipv4_lists/<STRING>/rules/<UINT16>/protocol

Description	Protocol name (IP, UDP or TCP)
Access	ro
Data Type	string
Release Version	1.16
CLI Example	show ipv4 access-lists <list name>
Notes	

/mlnxos/v1/vsr/<STRING>/acls/ipv4_lists/<STRING>/rules/<UINT16>/rule_type

Description	Rule type Format: "permit", "deny"
Access	ro
Data Type	string
Release Version	1.16
CLI Example	show ipv4 access-lists <list name>
Notes	

/mlnxos/v1/vsr/<STRING>/acls/ipv4_lists/<STRING>/rules/<UINT16>/sequence_num

Description	Rule sequence number
Access	ro
Data Type	uint16
Release Version	1.16
CLI Example	show ipv4 access-lists <list name>
Notes	

/mlnxos/v1/vsr/<STRING>/acls/ipv4_lists/<STRING>/rules/<UINT16>/source_ip

Description	Source IP address (Can be "any")
Access	ro
Data Type	string
Release Version	1.16
CLI Example	show ipv4 access-lists <list name>
Notes	

/mlnxos/v1/vsr/<STRING>/acls/ipv4_lists/<STRING>/rules/<UINT16>/source_ip_mask

Description	Source IP mask (Can be "any")
Access	ro
Data Type	string
Release Version	1.16
CLI Example	show ipv4 access-lists <list name>
Notes	

/mlnxos/v1/vsr/<STRING>/acls/ipv4_lists/<STRING>/rules/<UINT16>/source_port

Description	Source port number (can be "any")
Access	ro
Data Type	string
Release Version	1.16
CLI Example	show ipv4 access-lists <list name>
Notes	

/mlnxos/v1/vsr/<STRING>/acls/mac_lists/*

Description	List of MAC ACLs
Access	ro
Data Type	string
Release Version	1.116
CLI Example	show mac access-lists summary
Notes	

/mlnxos/v1/vsr/<STRING>/acls/mac_lists/add

Description	Add MAC ACLs			
Access	action			
Release Version	1.16			
Arguments	Name	Description	Data Type	Status
	acl_name	ACL name	string	mandatory
CLI Example	[no] mac access-list <mac-access-list-name>			
Notes				

/mlnxos/v1/vsr/<STRING>/acls/mac_lists/delete

Description	Delete MAC ACLs			
Access	action			
Release Version	1.16			
Arguments	Name	Description	Data Type	Status
	acl_name	ACL name	string	mandatory
CLI Example	[no] mac access-list <mac-access-list-name>			
Notes				

/mlnxos/v1/vsr/<STRING>/acls/mac_lists/<STRING>/rules/*

Description	List of MAC ACL rules
Access	ro
Data Type	uint16
Release Version	1.16
CLI Example	show mac access-lists summary
Notes	

/mlnxos/v1/vsr/<STRING>/acls/mac_lists/<STRING>/rules/add

Description	Add rule to the specified mac ACL			
Access	action			
Release Version	1.16			
Arguments	Name	Description	Data Type	Status
	rule_type	Rule type Format: "permit", "deny"	string	mandatory
	sequence_num	Rule sequence number	uint16	mandatory
	source_mac	Source MAC address (can be "any")	string	mandatory
	source_mac_mask	Source MAC mask (can be "any")	string	mandatory
	destination_-mac	Destination MAC address (can be "any")	string	mandatory
	destination_mac_mask	Destination MAC mask (can be "any")	string	mandatory
	protocol	Protocol number (can be "none")	uint8	mandatory
	cos_value	Class of service (can be "none")	uint8	mandatory
	acl_action_name	ACL action name (can be "none")	string	mandatory
	vlan_id	VLAN ID (can be "none")	int16	mandatory
CLI Example	[seq-number <sequence-number>] {deny|permit } {any | <source-mac> [mask <mac>]} {any |<destination-mac> [mask <mac>]} [protocol <protocol>] [cos <cos-value>] [vlan <vlan-id>] [action <action-id>]			
Notes				

/mlnxos/v1/vsr/<STRING>/acls/mac_lists/<STRING>/rules/delete

Description	Delete MAC ACL rule			
Access	action			
Release Version	1.16			
Arguments	Name	Description	Data Type	Status
	sequence_num	Rule sequence number	uint16	mandatory
CLI Example	mac access-list <list name> no <rule seq num>			
Notes				

/mlnxos/v1/vsr/<STRING>/acls/mac_lists/<STRING>/rules/<UINT16>/action_name

Description	ACL action name (can be "none")
Access	ro
Data Type	string
Release Version	1.16
CLI Example	show mac access-lists <list name>
Notes	

/mlnxos/v1/vsr/<STRING>/acls/mac_lists/<STRING>/rules/<UINT16>/cos_value

Description	Class of service (can be "none")
Access	ro
Data Type	string
Release Version	1.16
CLI Example	show mac access-lists <list name>
Notes	

/mlnxos/v1/vsr/<STRING>/acls/mac_lists/<STRING>/rules/<UINT16>/destination_mac

Description	Destination MAC address (can be "any")
Access	ro
Data Type	string
Release Version	1.16
CLI Example	show mac access-lists <list name>
Notes	

/mlnxos/v1/vsr/<STRING>/acls/mac_lists/<STRING>/rules/<UINT16>/destination_mac_mask

Description	Destination MAC mask (can be "any")
Access	ro
Data Type	string
Release Version	1.16
CLI Example	show mac access-lists <list name>
Notes	

/mlnxos/v1/vsr/<STRING>/acls/mac_lists/<STRING>/rules/<UINT16>/protocol

Description	Protocol number (can be "none")
Access	ro
Data Type	string
Release Version	1.16
CLI Example	show mac access-lists <list name>
Notes	

/mlnxos/v1/vsr/<STRING>/acls/mac_lists/<STRING>/rules/<UINT16>/rule_type

Description	Rule type Format: "permit", "deny"
Access	ro
Data Type	string
Release Version	1.16
CLI Example	show mac access-lists <list name>
Notes	

/mlnxos/v1/vsr/<STRING>/acls/mac_lists/<STRING>/rules/<UINT16>/sequence_num

Description	Rule sequence number
Access	ro
Data Type	uint16
Release Version	1.16
CLI Example	show mac access-lists <list name>
Notes	

/mlnxos/v1/vsr/<STRING>/acls/mac_lists/<STRING>/rules/<UINT16>/source_mac

Description	Source MAC address (can be "any")
Access	ro
Data Type	string
Release Version	1.16
CLI Example	show mac access-lists <list name>
Notes	

/mlnxos/v1/vsr/<STRING>/acls/mac_lists/<STRING>/rules/<UINT16>/source_mac_mask

Description	Source MAC mask (can be "any")
Access	ro
Data Type	string
Release Version	1.16
CLI Example	show mac access-lists <list name>
Notes	

/mlnxos/v1/vsr/<STRING>/acls/mac_lists/<STRING>/rules/<UINT16>/vlan_id

Description	VLAN ID (can be "none")
Access	ro
Data Type	string
Release Version	1.16
CLI Example	show mac access-lists <list name>
Notes	

/mlnxos/v1/vsr/<STRING>/acls/actions/*

Description	List of ACL actions
Access	ro
Data Type	string
Release Version	1.16
CLI Example	show access-list action [action-name] summary
Notes	

/mlnxos/v1/vsr/<STRING>/acls/actions/<STRING>/vlan_id

Description	VLAN identifier associated with action
Access	ro
Data Type	uint16
Release Version	1.16
CLI Example	show access-list [action name] summary
Notes	

2.5.2.2 LAGs

/mlnxos/v1/vsr/<STRING>/lags/lacp/enabled

Description	Enable/disable LACP
Access	rw
Data Type	bool
Release Version	1.16
CLI Example	[no] lacp
Notes	

/mlnxos/v1/vsr/<STRING>/lags/lacp/system_priority

Description	LACP system-priority
Access	rw
Data Type	uint32
Release Version	1.16
CLI Example	[no] lacp system-priority <1...65535>
Notes	

/mlnxos/v1/vsr/<STRING>/lags/load_balancing

Description	LAG load balancing method Format: "destination-ip", "destination-mac", "destination-port", "source-destination-ip", "source-destination-mac", "source-dest-port", "source-ip", "source-mac", "source-port"
Access	rw
Status	rw
Release Version	1.16
CLI Example	[no] port-channel load-balance ethernet <method>
Notes	

2.5.2.3 VLANs

/mlnxos/v1/vsr/<STRING>/vlans/*

Description	List of VLANs
Access	ro
Data Type	uint16
Release Version	1.13
CLI Example	show vlan
Notes	

/mlnxos/v1/vsr/<STRING>/vlans/add

Description	Add VLAN to list of VLANs			
Access	action			
Release Version	1.13			
Arguments	Name	Description	Data Type	Status
	vlan_id	VLAN ID	uint16	mandatory
CLI Example	[no] vlan {<vlan-id>, <vlan-range>}			
Notes				

/mlnxos/v1/vsr/<STRING>/vlans/delete

Description	Delete VLAN from list of VLANs			
Access	action			
Release Version	1.13			
Arguments	Name	Description	Data Type	Status
	vlan_id	VLAN ID	uint16	mandatory
CLI Example	[no] vlan {<vlan-id>, <vlan-range>}			
Notes				

/mlnxos/v1/vsr/<STRING>/vlans/<UINT16>/name

Description	VLAN name
Access	rw
Data Type	string
Release Version	1.13
CLI Example	vlan <vlan num> name <name>
Notes	

/mlnxos/v1/vsr/<STRING>/vlans/<UINT16>/vlan_if_index

Description	The index of the VLAN interface
Access	ro
Data Type	uint32
Release Version	1.16
CLI Example	N/A
Notes	

2.5.3 DCB

/mlnxos/v1/vsr/<STRING>/dcb/pfc/enabled

Description	Enables/disables PFC in the system and queries the current PFC state Allowed values: true/false
Access	rw
Data Type	bool
Release Version	1.18
CLI Example	[no] dcb priority-flow-control enable show dcb priority-flow-control
Notes	

/mlnxos/v1/vsr/<STRING>/dcb/pfc/priorities/*

Description	Lists PFC priorities
Access	ro
Data Type	Uint8
Release Version	1.18
CLI Example	
Notes	

/mlnxos/v1/vsr/<STRING>/dcb/pfc/priorities/<UINT8>/enabled

Description	Enables/Disables Per-priority PFC in the system and queries the Per-priority PFC state Allowed values: true/false
Access	rw
Data Type	bool
Release Version	1.18
CLI Example	[no] dcb priority-flow-control priority <prio> enable show dcb priority-flow-control
Notes	

2.6 Interface Configuration

2.6.1 General Interfaces

/mlnxos/v1/vsr/<STRING>/interfaces/*

Description	List of all interfaces in the system (identified by their interface index)
Access	ro
Data Type	uint16
Release Version	1.13
CLI Example	show interfaces ethernet <inf>
Notes	

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/delete

Description	Remove interface Valid for LAG and VLAN interfaces
Access	action
Data Type	N/A
Release Version	1.16
CLI Example	no interface port-channel <port channel num> no interface vlan <vlan-id>
Notes	

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/if_index

Description	Interface index
Access	ro
Data Type	uint16
Release Version	1.13
CLI Example	N/A
Notes	

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/type

Description	Interface type Format: "eth","lag","vlan","loopback"
Access	string
Data Type	active
Status	ro
Release Version	1.13
Notes	

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/physical_location

Description	Interface physical representation. For example: 1/5, Port-channel 30, L01/5
Access	ro
Data Type	string
Release Version	1.16
CLI Example	N/A
Notes	

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/enabled

Description	Interface administrative state
Access	rw
Data Type	bool
Release Version	1.16
CLI Example	[no] interface <if type> <if name> shutdown
Notes	This node is not available when switch is in IB router profile and the interface is not mapped to a subnet

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/description

Description	Query and configure the given interface description
Access	rw
Data Type	string
Release Version	1.17
CLI Example	interface <if type> <if name> description <description string> show interfaces <if type> <if name>
Notes	An empty value deletes the description of this interface

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/operational_state

Description	Query the given interface operational state
Access	ro
Data Type	string
Release Version	1.17
CLI Example	show interfaces <if type> <if name>
Notes	This node is not available when switch is in IB router profile and the interface is not mapped to a subnet

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/supported_speed

Description	Query maximum supported speed of the interface For example: InfiniBand lists "fdr, qdr, edr", and Ethernet lists "40000"
Access	ro
Data Type	string
Release Version	1.17
CLI Example	N/A
Notes	This node is not available when switch is in IB router profile and the interface is not mapped to a subnet

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/actual_speed

Description	Query actual current speed of the interface For example: InfiniBand lists "fdr, qdr, edr", and Ethernet gives the actual speed rate "40000"
Access	ro
Data Type	string
Release Version	1.17
CLI Example	show interfaces <if type> <if name>
Notes	This node is not available when switch is in IB router profile and the interface is not mapped to a subnet

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/configured_speed

Description	Configured speed of the interface For example: For InfiniBand it is a list of speed values such as "sdr", "qdr", and for Ethernet it can be "40000"
Access	ro
Data Type	string
Release Version	1.17
CLI Example	show interfaces <if type> <if name>
Notes	This node is not available when switch is in IB router profile and the interface is not mapped to a subnet

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/actual_rate

Description	Query the actual line rate of the port in MB For example: for InfiniBand, it is 40000, 56000; for Ethernet it is the same as actual speed
Access	ro
Data Type	string
Release Version	1.17
CLI Example	show interfaces <if type> <if name>
Notes	This node is not available when switch is in IB router profile and the interface is not mapped to a subnet

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/mtu

Description	Query the current MTU of the interface
Access	ro
Data Type	string
Release Version	1.17
CLI Example	show interfaces <if type> <if name>
Notes	This node is not available when switch is in IB router profile and the interface is not mapped to a subnet

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/supported_mtu

Description	Query the maximum supported MTU of the interface
Access	ro
Data Type	string
Release Version	1.17
CLI Example	show interfaces <if type> <if name>
Notes	This node is not available when switch is in IB router profile and the interface is not mapped to a subnet

2.6.2 Ethernet Interfaces

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/acls/ipv4/*

Description	List of bounded IPv4 ACLs
Access	string
Data Type	ro
Release Version	1.16
CLI Example	show access-lists summary
Notes	

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/acls/ipv4/add

Description	Bind an IPv4 ACL to the specified interface			
Access	action			
Release Version	1.16			
Arguments	Name	Description	Data Type	Status
	acl_name	ACL name	string	mandatory
CLI Example	interface ethernet <if name> ipv4 port access-group <list name>			
Notes				

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/acls/ipv4/delete

Description	Remove binding of interface to an ipv4 ACL			
Access	action			
Release Version	1.16			
Arguments	Name	Description	Data Type	Status
	acl_name	ACL name	string	mandatory
CLI Example	no interface ethernet <if name> ipv4 port access-group <list name>			
Notes				

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/acls/mac/*

Description	List of bounded MAC ACLs
Access	ro
Data Type	string
Release Version	1.16
CLI Example	show access-lists summary
Notes	

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/acls/mac/add

Description	Bind a MAC ACL to the specified interface			
Access	action			
Release Version	1.16			
Arguments	Name	Description	Data Type	Status
	acl_name	ACL name	string	mandatory
CLI Example	interface ethernet <ifname> mac port access-group <list name>			
Notes				

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/acls/mac/delete

Description	Remove binding of interface to a MAC ACL			
Access	action			
Release Version	1.16			
Arguments	Name	Description	Data Type	Status
	acl_name	ACL name	string	mandatory
CLI Example	no interface ethernet <if name> mac port access-group <list name>			
Notes				

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/dcb/pfc/admin_mode

Description	Sets PFC admin mode on the given interface Allowed values: “On” or “Off” or an empty string to return to the default value
Access	rw
Data Type	string
Release Version	1.18
CLI Example	[no] interface <if type> <if name> dcb priority-flow-control mode <on|off> show dcb priority-flow-control
Notes	

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/counters/xmit_wait

Description	xmit_wait counter
Access	ro
Data Type	uint64
Release Version	1.16
CLI Example	show interface ethernet <inf> counters
Notes	

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/ipv4/ip_address

Description	Interface IP address(Valid for VLAN interfaces)
Access	ro
Data Type	ipv4addr
Release Version	1.16
CLI Example	interface vlan <vlan num> ip address <ip address> <ip mask>
Notes	

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/ipv4/net_mask

Description	Interface network mask(Valid for VLAN interfaces)
Access	ro
Data Type	ipv4addr
Release Version	1.16
CLI Example	interface vlan <vlan num> ip address <ip address> <ip mask>
Notes	

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/ipv4/set_address

Description	Set IP address and network mask			
Access	action			
Release Version	1.16			
Arguments	Name	Description	Data Type	Status
	ip_address	IP Address	ipv4addr	mandatory
	net_mask	Network mask	ipv4addr	mandatory
CLI Example	interface vlan <vlan num> ip address <ip> <net mask>			
Notes	Relevant for VLAN interface only			

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/ipv4/ospf/area

Description	Configure or query the OSPF area for the given interface
Access	rw
Data Type	Configuration: uint16 Querying: string
Release Version	1.16
CLI Example	interface vlan <vlan num> ip address <ip> <net mask> show ip ospf
Notes	Empty value or -1 means delete this configuration When querying this node, the value returned is a string of the area IP address L3 must be enabled on the system to operate OSPF features. See the command “ip l3” for more details

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/lag/membership

Description	LAG Interface which this interface is member of.
Access	ro
Data Type	uint16
Release Version	1.16
CLI Example	show interfaces port-channel summary
Notes	If not LAG member, the value is zero

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/lag/lag_mode

Description	LAG Interface's mode Format: "active", "passive", "on"
Access	ro
Data Type	string
Release Version	1.16
CLI Example	show interfaces port-channel summary
Notes	Valid only for LAG interfaces

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/priorities/*

Description	Lists priorities within the interface
Access	ro
Data Type	Uint8
Release Version	1.18
CLI Example	
Notes	

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/priorities/<UINT8>/traffic_class

Description	Priority mapping (QoS) for interface—maps priority to traffic class Format: -1 means to use default mapping, other values indicate traffic class
Access	rw
Data Type	int32
Release Version	1.18
CLI Example	[no] interface <if type> <if name> vlan map-priority <priority> traffic-class <tc> show dcb ets interface <if type> <port num>
Notes	

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/vlans/allowed/*

Description	List of allowed VLANs on the interface (Valid in case VLAN mode is either "trunk" or "hybrid")
Access	ro
Data Type	uint16
Release Version	1.13
CLI Example	show interfaces switchport
Notes	

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/vlans/allowed/add

Description	Add VLANs to list of allowed VLANs on the interface (Valid in case vlan mode is either "trunk" or "hybrid")			
Access	action			
Release Version	1.13			
Arguments	Name	Description	Data Type	Status
	vlan_ids	Single VLAN ID (e.g. "5") or VLAN range (e.g. "4-7")	string	mandatory
CLI Example	interface ethernet <if name> switchport {trunk | hybrid} allowed vlan add <vlan-range or valn num>			
Notes				

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/vlans/allowed/delete

Description	Delete VLANs from list of allowed VLANs on the interface (Valid in case VLAN mode is either "trunk" or "hybrid")			
Access	action			
Release Version	1.13			
Arguments	Name	Description	Data Type	Status
	vlan_ids	Single VLAN ID (e.g. "5") or VLAN range (e.g. "4-7")	string	mandatory
CLI Example	interface ethernet <if name> switchport <trunk|hybrid> allowed-vlan remove <vlan num or range>			
Notes				

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/vlans/mode

Description	Interface's VLAN mode Format: "access", "trunk", "hybrid", "access-dcb"
Access	rw
Data Type	string
Release Version	1.13
CLI Example	interface ethernet <if-name> switchport mode <access | trunk | hybrid | access-dcb> show interfaces switchport
Notes	

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/vlans/pvid

Description	The access (native) VLAN configured on the interface (Not valid in case vlan mode is "trunk")
Access	uint16
Data Type	active
Status	rw
Release Version	1.13
CLI Example	interface ethernet <if-name> switchport access vlan <vlan-id> show interfaces switchport
Notes	

/mlnxos/v1/vsr/<STRING>/interfaces/add_lag

Description	Add a LAG to the system			
Access	action			
Release Version	1.16			
Arguments	Name	Description	Data Type	Status
	if_index	Interface index for LAG	uint16	mandatory
CLI Example	[no] interface port-channel <1...4096>			
Notes				

/mlnxos/v1/vsr/<STRING>/interfaces/add_vlan_interface

Description	Add VLAN interface Returns generated if_index as return value		
Access	action		
Release Version	1.16		
Arguments	Name	Description	Data Type
	vlan_id	VLAN ID	uint16
CLI Example	[no] interface vlan <vlan-id>		
Notes			

2.6.3 InfiniBand Interfaces

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/supported_width

Description	Query maximum supported width of the interface Format: list of "1X ,4X"
Access	ro
Data Type	string
Release Version	1.17
CLI Example	show interfaces <if type> <if name>
Notes	This node is not available when switch is in IB router profile and the interface is not mapped to a subnet

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/actual_width

Description	Query actual current width of the interface (InfiniBand) Format: "1X","4X"
Access	ro
Data Type	string
Release Version	1.17
CLI Example	show interfaces <if type> <if name>
Notes	This node is not available when switch is in IB router profile and the interface is not mapped to a subnet

/mlnxos/v1/vsr/<STRING>/interfaces/<UINT32>/configured_width

Description	Query configured width of the interface Format: a list of width values (e.g. "1X, 4X")
Access	ro
Data Type	string
Release Version	1.17
CLI Example	show interfaces <if type> <if name>
Notes	This node is not available when switch is in IB router profile and the interface is not mapped to a subnet

2.7 Legacy API Nodes

2.7.1 General

/mlnxos/v1/chassis/reset

Description	Reset the device
Access	action
Data Type	N/A
Release Version	1.0
CLI Example	reload force
Notes	

2.7.2 File Transfer

/mlnxos/v1/chassis/file_transfer/download

Description	Download file to device			
Access	action			
Release Version	1.0			
Arguments	Name	Description	Data Type	Status
	remote_url	URL to file's location (Should contain username & password).Example: "scp://user:password@server/path/file.img"	uri	mandatory
	local_filename	Local filename to use when downloading. Example: "temp.img"	string	mandatory
	type	Type of file.Format: "img", "db"	string	mandatory
CLI Example	image fetch <URL> [<filename>] configuration fetch <uri>			
Notes				

/mlnxos/v1/chassis/file_transfer/upload

Description	Upload file from device			
Access	action			
Release Version	1.0			
Arguments	Name	Description	Data Type	Status
	remote_url	URL to file target location (Should contain username & password).Example: "scp://user:password@server/path/file.img"	uri	mandatory
	type	Type of file.Format: "db"	string	mandatory
CLI Example	configuration upload <filename> <uri>			
Notes				

/mlnxos/v1/chassis/file_transfer/state/download/percent_done

Description	Percent done when downloading (progress tracking)
Access	ro
Data Type	float32
Release Version	1.0
CLI Example	N/A
Notes	

/mlnxos/v1/chassis/file_transfer/state/upload/percent_done

Description	Percent done when uploading (progress tracking)
Access	ro
Data Type	float32
Release Version	1.0
CLI Example	N/A
Notes	

/mlnxos/v1/chassis/file_transfer/state/download/response_code

Description	Response error code for download operation
Access	ro
Data Type	uint32
Release Version	1.0
CLI Example	N/A
Notes	

/mlnxos/v1/chassis/file_transfer/state/upload/response_code

Description	Response error code for upload operation
Access	ro
Data Type	uint32
Release Version	1.0
CLI Example	N/A
Notes	

/mlnxos/v1/chassis/file_transfer/state/download/response_msg

Description	Response error message for download operation
Access	ro
Data Type	string
Release Version	1.0
CLI Example	N/A
Notes	

/mlnxos/v1/chassis/file_transfer/state/upload/response_msg

Description	Response error message for upload operation
Access	ro
Data Type	string
Release Version	1.0
CLI Example	N/A
Notes	

/mlnxos/v1/chassis/file_transfer/state/download/state

Description	State of download operation Format: "idle", "running", "completed"
Access	ro
Data Type	string
Release Version	1.0
CLI Example	N/A
Notes	

/mlnxos/v1/chassis/file_transfer/state/upload/state

Description	State of upload operation Format: "idle", "running", "completed"
Access	ro
Data Type	string
Release Version	1.0
CLI Example	N/A
Notes	

2.7.3 Image Management

/mlnxos/v1/chassis/image/install

Description	Install image			
Access	action			
Release Version	1.0			
Arguments	Name	Description	Data Type	Status
	image_name	Name of image file to install	string	mandatory
CLI Example	image install <image name>			
Notes				

/mlnxos/v1/chassis/image/state/percent_done

Description	Percent done when installing image (progress tracking)
Access	ro
Data Type	float32
Release Version	1.0
CLI Example	N/A
Notes	

/mlnxos/v1/chassis/image/state/response_code

Description	Response error code for image install operation
Access	ro
Data Type	uint32
Release Version	1.0
CLI Example	N/A
Notes	

/mlnxos/v1/chassis/image/state/response_msg

Description	Response error message for image install operation
Access	ro
Data Type	string
Release Version	1.0
CLI Example	N/A
Notes	

/mlnxos/v1/chassis/image/state/state

Description	State of image install operation Format: "idle", "running", "completed"
Access	ro
Data Type	string
Release Version	1.0
CLI Example	N/A
Notes	