API Reference for Catalyst 1200 and 1300 Switches

Table of Contents

1. Introduction

2. APIs

2.1 Login/Logout Operations

2.1.1 Login

2.1.2 Logout

2.2 Get Operations

2.2.1 Queries

2.2.2 Filters

2.2.3 Additional Parameters

2.2.4 Request for Non-existing or Non-supported Sections

2.3 Post Operations

2.3.1 Action Set

2.3.2 Actions Delete and Delete All

2.3.3 Actions Restore and RestoreAll

2.4 Section "VLANList"

2.4.1 Tag "VLANID"

2.4.2 Tag "VLANName"

2.4.3 Tag "authorizationType"

2.4.4 Tag "VLANType"

2.5 Section "Standard802_3List"

2.5.1 Tag "interfaceName"

2.5.2 Tag "interfaceType"

2.5.3 Tag " interfaceID"

2.5.4 Tag " interfaceDescription"

2.5.5 Tag " MACAddress"

2.5.6 Tag " adminState"

2.5.7 Tag " linkState"

2.5.8 Tag " linkStateTrapEnabled"

2.5.9 Tag " mediaType"

2.5.10 Tag " adminComboMode"

2.5.11 Tag " speedAdmin"

2.5.12 Tag " speedOper"

2.5.13 Tag " duplexOperMode"

2.5.14 Tag " duplexOperModeDot3"

2.5.15 Tag " duplexAdminMode"

2.5.16 Tag " autoNegotiationOperEnabled"

2.5.17 Tag " autoNegotiationAdminEnabled"

2.5.18 Tag " backPressureAdminEnabled"

2.5.19 Tag " backPressureOperEnabled"

2.5.20 Tag " flowControlAdminType"

2.5.21 Tag " flowControlOperType"

2.5.22 Tag " adminAdvertisementList"

2.5.23 Tag " operAdvertisementList"

2.5.24 Tag " MDIAdminMode"

2.5.25 Tag " MDIOperMode"

2.5.26 Tag " reActivate"

2.5.27 Tag " LACPEnabled"

2.5.28 Tag " LAGID"

2.5.29 Tag "interfaceName"

2.5.30 Tag "interfaceID"

2.5.31 Tag "interfaceType"

2.6 Section "IPv4InterfaceList"

2.6.1 Tag "IPAddr"

2.6.2 Tag "interfaceName"

2.6.3 Tag "subnetMask"

2.6.4 Tag "owner"

1. Introduction

Catalyst 1300/1200 switches provide XML Management Interface for network management applications (NMAs). Interaction between an NMA and an embedded XML engine is carried out over the HTTP/HTTPS protocol. Sending XML files via HTTP Post enables the NMA to configure a device.

Receiving XML files via HTTP Get enables the NMA to obtain current information about the state ofa device.

Three types of XML files are used in the context of XML management:

- Post-XML is used for sending configuration data to a device
- Get-XML is used for reading configuration data from a device
- Status-XML is used to inform the NMA about the results of a previous operation.

Post and Get-XML files may contain one or more XML sections, which are called XML APIs.

An XML section definition includes the follow elements:

- List of supported actions (Action List table)
- Section structure (Section Structure table)
- List of available parameters - Queries, Filters and Additional Parameters ("Section Description" table)
- Nodes definitions.

An XML section can include nodes of the following types:

- Node-container containing other nodes
- Data-node containing data value only

The set of the following attributes is provided for each data-node (Data Nodes and Tag

Attributes tables):

- Access (RW/RO)
- Key (Yes/No)
- Mandatory (Yes/No)
- MIB Name
- Data Type (see Table 1")
- Default value
- Min Value
- Max Value

Default and min/max values may be model-dependent. For String and Octet String types, the min/max values indicate the min/max string length. A node definition may be supplemented by an Enumerator List or Bitmap List.

2. APIs

This section will cover the major components APIs .

2.1 Login/Logout Operations

2.1.1 Login

To open an HTTP session, the NMA must send an HTTP Request to the device in the following format:

https://x.x.x.x/System.xml?action=login&user=cisco&password=cisco

The device returns the authentication result in a Status-XML containing the statusCode and statusString nodes.

<?xml version='1.0' encoding='UTF-8'?>

<ResponseData>

<ActionStatus>

<version>1.0</version>

<requestURL>/System.xml</requestURL>

<statusCode>0</statusCode>

<deviceStatusCode>0</deviceStatusCode>

<statusString>OK</statusString>

</ActionStatus>

</ResponseData>

2.1.2 Logout

To close the HTTP session, the NMA must send the following request:

https://x.x.x.x/System.xml?action=logout

2.2 Get Operations

To obtain the data, the NMA sends an HTTP Request (URL) to the device in a specific format. The URL format enables the use of filters, queries and additional parameters to optimize the traffic between the device and the NMA.

The following is the URL format:

wcd?{Section[&Query] [&Special Parameters] [&Filter]}...[{Section[&...] }]:

where ***Section*** is the Name of Section according to the XML spec.

The URL format enables retrieving one or more blocks of data within the same GET XML.

As can be seen in the following example, the <AutoSmartPortGlobalSetting> and <CDPGlobalSetting> sections are the children nodes of <DeviceConfiguration> node.

https://x.x.x.x/wcd?{AutoSmartPortGlobalSetting}{CDPGlobalSetting}

<?xml version=" 1.0" encoding=" UTF-8" ?>

<ResponseData>

<DeviceConfiguration>

<version>1.0</version>

<AutoSmartPortGlobalSetting type=" section" >

<statusAdmin>3</statusAdmin>

<statusOper>2</statusOper>

<voiceVLANOperStatus>0</voiceVLANOperStatus>

<learningProtocol>11</learningProtocol>

</AutoSmartPortGlobalSetting>

<CDPGlobalSetting type=" section" >

<enbl>1</enbl>

<applianceTLVAdvertisement>1 </applianceTLVAdvertisement>

<CDPDUMode>2</CDPDUMode>

<versionAdvertised>2</versionAdvertised>

<mandatoryTLVValidation>2

</mandatoryTLVValidation>

<messageInterval>60</messageInterval>

<holdTime>180</holdTime>

<deviceIdFormat>2</deviceIdFormat>

<sourceInterfaceName>none</sourceInterfaceName>

<logMismatchDuplex>1</logMismatchDuplex>

<logMismatchVoiceVLAN>1</logMismatchVoiceVLAN>

<logMismatchNativeVLAN>1</logMismatchNativeVLAN>

</CDPGlobalSetting>

</DeviceConfiguration>

<ActionStatus>

<requestURL>CDPGlobalSetting</requestURL>

<statusCode></statusCode>

</ActionStatus>

</ResponseData>

2.2.1 Queries

A ***Query*** is a logical expression in the following format:

tagName=Value[&tagName=Value]

The query searches for a unique entry, identified by a set of key values. It is the most effective method of searching for data, since it uses direct access instead of full table scan.

A single query always returns a single record. The query format is a list of key-value pairs delimited by the "&" symbol. The query must contain values for all tags specified as the key in given XML Scheme.

In the following sample, the NMA retrieves a row for a single key:

https://x.x.x.x/wcd?{AdminUserSetting&userName=a1}

In the following sample, the NMA retrieves a row for the three keys:

https://x.x.x.x/wcd?{ForwardingStaticTable&VLANID=1&MACAddress=00:3f:bd:45:5a:b2&interfaceName=fa2}

2.2.2 Filters

A ***Filter*** is a set of Boolean predicates combined by the operation "||" and "&&" in a logical expression with following format:

filter=[(] (predicate1) [&& | || (predicate2) ...)],

where the predicate is the following expression:

nodeName = | != | > | < | >= |<= value

The filter searches for a group of entries that satisfy some logical condition. This condition is a logical expression, which can include XML tags. A set of these tags is defined per XML section in the Scheme description. Filter is the slowest way to find data, because it causes a full table scan.

In the following sample, an NMA retrieves only static entries from the Forwarding database:

https://x.x.x.x/wcd?{ForwardingTable&filter=(addressType=5)}

> **Note:**
> String values presented in the filter expression must be enclosed in quotes. This is not required for query expressions:

https://x.x.x.x0/wcd?{ForwardingTable&filter=(MACAddress='00:01:b0:00:00:00')}

Related Get-XML:

<?xml version=" 1.0" encoding=" UTF-8" ?>

<ResponseData>

<DeviceConfiguration>

<version>1.0</version>

<ForwardingTable type=" section" >

<Entry>

<VLANName>default</VLANName>

<VLANID>1</VLANID>

<MACAddress>00:01:b0:00:00:00</MACAddress>

<interfaceType>0</interfaceType>

<interfaceName>none</interfaceName>

<addressType>4</addressType>

</Entry>

</ForwardingTable>

</DeviceConfiguration>

<ActionStatus>

<requestURL>ForwardingTable</requestURL>

<statusCode />

</ActionStatus>

</ResponseData>

2.2.3 Additional Parameters

The Additional Parameters extend the functionality of the XML Scheme. Applying additional parameters is similar to applying macro-commands. As well as macro-commands, a set of parameters should be predefined for each section. Processing of parameters similar to macro-commands sometimes requires several operations. The format of additional parameters is the same as a query (parameter-value pairs, separated by the symbol &), although the result of applying additional parameters might consist of a group of records in the same way as the result of filtering. The difference between additional parameters and filtering is that the processing of additional parameters does not cause a full table scan.

> **Note:**
> Additional parameters are not required to be XML nodes belonging to the section. They may be presented in URL only, as shown in the example of using unitID described below.

The most frequently additional parameters used are the following:

- **interfaceType** – is used to filter the tables containing interfaces (Ports, LAGs, VLANs). In the following example retrieves all rows relating to link aggregation interfaces (LAG):

  https://x.x.x.x/wcd?{Standard802_3List&interfaceType=2}

  (See Enumerator List for "interfaceType" node in "Standard802_3List" section description).

- **unitID** – is used to filter information related to the selected unit in stackable device. In the following example NMA retrieves all rows relating to the ports belonged to the first unit:

  https://x.x.x.x/wcd?{Standard802_3List&interfaceType=1&unitID=1}

- **count** – is used to specify the number of rows that satisfy the filter to be sent to NMA. In the following example retrieves three first rows relating to the link aggregation interfaces (LAG):

  https://x.x.x.x/wcd?{Standard802_3List&interfaceType=2&count=3}

- **entryCount** – is used to obtain the total number of entries in the table in the special <entryCount> node:

https://x.x.x.x0/wcd?{ForwardingTable&entryCount=true}

<?xml version="1.0" encoding="UTF-8" ?>

<ResponseData>

<DeviceConfiguration>

<version>1.0</version>

<ForwardingStaticTable type="section" >

<entryCount>3</entryCount>

<Entry>

<VLANName>default</VLANName>

<VLANID>1</VLANID>

<MACAddress>00:3f:bd:45:5a:b1</MACAddress>

<interfaceType>1</interfaceType>

<limitationMode>1</limitationMode>

<interfaceName>fa1/0/20</interfaceName>

<addressStatus>3</addressStatus>

</Entry>

<Entry>

<VLANName>default</VLANName>

<VLANID>1</VLANID>

<MACAddress>00:3f:bd:45:5a:b2</MACAddress>

<interfaceType>1</interfaceType>

<limitationMode>1</limitationMode>

<interfaceName>fa1/0/21</interfaceName>

<addressStatus>4</addressStatus>

</Entry>

<Entry>

<VLANName>default</VLANName>

<VLANID>1</VLANID>

<MACAddress>00:3f:bd:45:5a:b3</MACAddress>

<interfaceType>1</interfaceType>

<limitationMode>1</limitationMode>

<interfaceName>fa1/0/22</interfaceName>

<addressStatus>5</addressStatus>

</Entry>

</ForwardingStaticTable>

</DeviceConfiguration>

<ActionStatus>

<requestURL>ForwardingStaticTable</requestURL>

<statusCode></statusCode>

</ActionStatus>

</ResponseData>

2.2.4 Request for Non-existing or Non-supported Sections

2.2.4.1 Non-existing Sections

If a NMA requests a non-existing scheme, it receives the following Status-XML:

https://x.x.x.x0/wcd?{XXX}

<?xml version=" 1.0" encoding=" UTF-8" ?>

<ResponseData>

<DeviceConfiguration>

<version>1.0</version>

</DeviceConfiguration>

<ActionStatus>

<requestURL>XXX</requestURL>

<statusCode>-1</statusCode>

<statusString>Cannot find the section XXX !</statusString>

</ActionStatus>

</ResponseData>

2.2.4.2 Non-supported Sections

If a NMA requests an API not supported in specific SKU, it receives the following Status-XML:

<?xml version=" 1.0" encoding=" UTF-8" ?>

<ResponseData>

<DeviceConfiguration>

<version>1.0</version>

</DeviceConfiguration>

<ActionStatus>

<requestURL>SNMPGlobalSetting</requestURL>

<statusCode>3</statusCode>

<statusString>Feature is not supported</statusString>

</ActionStatus>

</ResponseData>

2.3 Post Operations

To configure a device the NMS should to specify the “/wcd” URI in the HTTP Header and put the POST-XML in the HTTP Body.

The HTTP POST example is shown below:

POST /wcd HTTP/1.0

Connection: keep-alive

Content-Length: 220

EzVersion: 1.0

QueryString: action=set

sessionID: UserId=10.5.70.82&5581720&;path=/

Host: x.x.x.x0

Accept: text/html, */*

Accept-Encoding: identity

User-Agent: Mozilla/3.0 (compatible; Indy Library)

<?xml version='1.0' encoding='utf-8'?>

<DeviceConfiguration>

<GlobalActions action=" set" >

<factoryDefault/>

</GlobalActions>

<Reload action=" set" />

</DeviceConfiguration>

The Post-XML might contain one or more XML Sections with action attribute (set | delete | restore | etc). The POST XML, shown above, includes two sections “GlobalActions” and “Reload” which cause the device to restore factory defaults and then to reboot. As in Get XML, all section nodes are included in the <DeviceConfiguration> container.

After each Post operation, the NMA receives an HTTP response containing the Status-XML with a return code (statusCode tag) and a diagnostics message (statusString tag), referring to the last operation

<?xml version='1.0' encoding='UTF-8'?>

<ResponseData>

<ActionStatus>

<version>1.0</version>

<requestURL>wcd</requestURL>

<statusCode>0</statusCode>

<deviceStatusCode>0</deviceStatusCode>

<statusString>OK</statusString>

<localMsg>

<msgId></msgId>

</localMsg>

</ActionStatus>

</ResponseData>

If localization is enabled on the device (on Nikola devices, the localization is always enabled), the Status-XML includes a unique identifier of the diagnostics message (see container <localMsg>) and the list of parameters (see tag <arg>):

<?xml version='1.0' encoding='UTF-8'?>

<ResponseData>

<ActionStatus>

<version>1.0</version>

<requestURL>wcd</requestURL>

<statusCode>3</statusCode>

<deviceStatusCode>0</deviceStatusCode>

<statusString>VLAN 2000: Port g5 cannot be VLAN Member and Forbidden.</statusString>

<localMsg>

<msgId>vlanPortEgressAndForbidden</msgId>

<arg>2000</arg>

<arg>g5</arg>

</localMsg>

</ActionStatus>

</ResponseData>

It is important to note that transaction mechanism is not supported for Post operations. Executing Post -XML suspended on the first failed operation, the NMA receives the relevant Status-XML and the rest of the operations remain unfulfilled.

2.3.1 Action Set

The Set action is used for updating MIB scalars and updating or adding entries in MIB tables.

Post-XML must contain all keys and mandatory data nodes. Read-only nodes are ignored.

Selecting which operation to execute (add or update) depends on the API being used. For some APIs, the updating is performed by deleting and inserting new values only. At any rate, the XML API behavior is consistent with the behavior of the corresponding CLI commands.

In the following example, the NMA adds two new users to the SNMP user list:

<?xml version=" 1.0" encoding='utf-8'?>

<DeviceConfiguration>

<SNMPUserList action=" set" >

<SNMPUserEntry>

<userName>user_a1</userName>

<SNMPGroupName>group</SNMPGroupName>

<remoteEngineID>987654321a</remoteEngineID>

<authenticationMethod>1</authenticationMethod>

<password>12345678901234567890123456789012</password>

```
</SNMPUserEntry>

<SNMPUserEntry>

<userName>user_a31</userName>

<SNMPGroupName>group</SNMPGroupName>

<remoteEngineID>987654321a</remoteEngineID>

<authenticationMethod>3</authenticationMethod>
<SNMPAuthenticationKey>12345678901234567890123456789012

</SNMPAuthenticationKey>

</SNMPUserEntry>

</SNMPUserList>

</DeviceConfiguration>
```

2.3.2 Actions Delete and Delete All

The POST XML with a delete action must contain all keys to identify the entries to be deleted. The deleteAll action always includes only one section node, since all entries should be removed from the table.

In the following example, the NMA deletes user user_a1 from SNMP User List:

```
<?xml version=" 1.0"  encoding='utf-8'?>

<DeviceConfiguration>

<SNMPUserList action=" delete" >

<SNMPUserEntry>

<userName>user_a1</userName>

<remoteEngineID>987654321a</remoteEngineID>

</SNMPUserEntry>

</SNMPUserList>

</DeviceConfiguration>
```

In following example, the NMA deletes all users from the SNMP User List:

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

<DeviceConfiguration>

<SNMPUserList action=" deleteAll" />

</DeviceConfiguration>

2.3.3 Actions Restore and RestoreAll

The POST XML with the restore action must contain all keys to identify the entries to be restored. In addition, each entry must include the tags to indicate the values to be restored. The restoreAll action includes only a single section node, since each tag from all entries is restored.

In the following example, the NMA restores to default the SNMP Local Engine ID value:

<?xml version=" 1.0" encoding='utf-8'?>

<DeviceConfiguration>

<SNMPGlobalSetting action=" restore" >

<localEngineID/>

</SNMPGlobalSetting>

</DeviceConfiguration>

In the following example, the NMA resets the initial state for all nodes belonging to SNMPGlobalSetting Section:

<?xml version=" 1.0" encoding='utf-8'?>

<DeviceConfiguration>

<SNMPGlobalSetting action=" restoreAll" />

</DeviceConfiguration>

2.4 Section " VLANList"

Table Action List

get	Retrieve VLAN(s) properties without members
set	Add or update VLAN(s),

	VLANType can be set from dynamicGvrp to permanent only
delete	Delete only permanent VLAN(s)
deleteAll	Delete all permanent VLANs

Table Section Structure

XML Structure
<VLANList action=" ?" > <VLAN> <VLANID/> <VLANName/> <authorizationType/> <VLANType/> </VLAN> </VLANList>

Table Section Description

Description
Query: VLANID Filter: VLANName, VLANType

Param: count

Table Data Nodes

Tag Name	MIB Name	Access	Key	Mandatory
VLANID	dot1qVlanIndex	R/W	Yes	Yes
VLANName	dot1qVlanStaticName	R/W	No	No
authorizationType	rldot1xUnAuthenticatedVlanTable, rldot1xGuestVlanVID	R/O	No	No
VLANType	dot1qVlanStatus	R/O	No	No

2.4.1 Tag "VLANID"

Table Tag Attributes

Path	/DeviceConfiguration/VLANSetting/VLANList/VLAN/VLANID
Description	Max value depends on max number of VLANs in the device

Type	Integer
Default Value	1
Min Value	1
Max Value	4094

2.4.2 Tag "VLANName"

Table Tag Attributes

Path	/DeviceConfiguration/VLANSetting/VLANList/VLAN/VLANName
Description	An administratively assigned string, which may be used to identify the VLAN.
Type	String
Default Value	N/A
Min Value	0
Max Value	32

2.4.3 Tag "authorizationType"

Table Tag Attributes

Path	/DeviceConfiguration/VLANSetting/VLANList/VLAN/authorizationType
Description	Tag value should be calculated as in CLI. CLI command: show vlan [id vlan-id | name vlan-name]
Type	Integer
Default Value	N/A
Min Value	1
Max Value	3

Table Enumerator List

Required	1
Not Required	2
Guest	3

2.4.4 Tag "VLANType"

Table Tag Attributes

Path	/DeviceConfiguration/VLANSetting/VLANList/VLAN/VLANType
Description	This object indicates the status of this entry. other(1) – this entry is currently in use but the conditions under which it will remain so differ from the following values. permanent(2) – this entry, corresponding to an entry in dot1qVlanStaticTable, is currently in use and will remain so after the next reset of the device. The port lists for this entry include ports from the equivalent dot1qVlanStaticTable entry and ports learnt dynamically. dynamicGvrp(3) – this entry is currently in use and will remain so until removed by GVRP. There is no static entry for this VLAN and it will be removed when the last port leaves the VLAN. VLANType can be set ONLY from dynamicGvrp to permanent

Type	Integer
Default Value	2
Min Value	1
Max Value	3

Table Enumerator List

other	1
permanent	2
dynamicGvrp	3

2.5 Section "Standard802_3List"

Table Action List

get	Get entry (entries)
set	Update specified fields of entry (entries)
restore	Restore defaults of specified fields of entry (entries)

restoreAll	Restore defaults of all fields of all entries

Table Section Structure

XML Structure
<Standard802_3List action=" ?" > <Entry> <interfaceName/> <interfaceType/> <interfaceID/> <interfaceDescription/> <MACAddress/> <adminState/> <linkState/> <linkStateTrapEnabled/> <mediaType/> <adminComboMode/> <speedAdmin/> <speedOper/> <duplexOperMode/> <duplexOperModeDot3/> <duplexAdminMode/> <autoNegotiationOperEnabled/>

```
<autoNegotiationAdminEnabled/>
<backPressureAdminEnabled/>
<backPressureOperEnabled/>
<flowControlAdminType/>
<flowControlOperType/>
<adminAdvertisementList/>
<operAdvertisementList/>
<MDIAdminMode/>
<MDIOperMode/>
<reActivate/>
<LACPEnabled/>
<LAGID/>
<PVEGroupUplink>
<interfaceName/>
<interfaceID/>
<interfaceType/>
</PVEGroupUplink>
</Entry>
</Standard802_3List>
```

Table Section Description

Description
CLI: show interfaces advertise, show interfaces configuration, show

interfaces description, show interfaces status

Query: interfaceName

Filter: any field except LACPEnabled, LAGID, and inner tags of PVEGroupUplink

Param: interfaceType, unitID, count

Table Data Nodes

Tag Name	MIB Name	Access	Key	Mandatory
interface Name	ifIndex	R/W	Yes	Yes
interface Type	ifIndex	R/O	No	No
interface ID	ifIndex	R/O	No	No
interface Description	ifAlias	R/W	No	No
MACAddress	ifPhysAddress	R/O	No	No
adminState	ifAdminStatus	R/W	No	No
linkState	ifOperSt	R/O	No	No

	atus			
linkStateTrapEnabled	ifLinkUpDownTrapEnable	R/W	No	No
mediaType	swIfTransceiverType	R/O	No	No
adminComboMode	swIfAdminComboMode	R/O	No	No
speedAdmin	swIfSpeedAdminMode	R/W	No	No
speedOper	ifHighSpeed	R/O	No	No
duplexOperMode	swIfDuplexOperMode	R/O	No	No
duplexOperModeDot3	dot3StatsDuplexStatus	R/O	No	No
duplexAdminMode	swIfDuplexAdminMode	R/W	No	No

autoNegotiationOperEnabled	swIfOperSpeedDuplexAutoNegotiation	R/O	No	No
autoNegotiationAdminEnabled	swIfSpeedDuplexAutoNegotiation	R/W	No	No
backPressureAdminEnabled	swIfBackPressureMode	R/W	No	No
backPressureOperEnabled	swIfOperBackPressureMode	R/O	No	No
flowControlAdminType	swIfFlowControlMode	R/W	No	No
flowControlOperType	swIfOperFlowControlMode	R/O	No	No
adminAdvertisementList	swIfAdminSpeedDuplexAutoNegotiationLocalCapabilities	R/W	No	No

operAdvertisementList	swIfOperSpeedDuplexAutoNegotiationLocalCapabilities	R/O	No	No
MDIAdminMode	swIfAdminMdix	R/W	No	No
MDIOperMode	swIfOperMdix	R/O	No	No
reActivate	swIfReActivate	R/W	No	No
LACPEnabled	dot3adAggPortActorAdminState:lacpActivity(0)	R/W	No	No
LAGID	dot3adAggPortActorAdminKey	R/W	No	No
interface Name	ifIndex	R/W	No	No
interface ID	ifIndex	R/O	No	No
interface	ifIndex	R/O	No	No

Type				

2.5.1 Tag "interfaceName"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/interfaceName
Description	The unique textual name of the interface.
Type	String
Default Value	N/A
Min Value	1
Max Value	3

2.5.2 Tag "interfaceType"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/interfaceType
Description	The type of the interface.

Type	Integer
Default Value	N/A
Min Value	1
Max Value	2

Table Enumerator List

port	1
LAG	2

2.5.3 Tag "interfaceID"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/interfaceID
Description	Obsolete.
Type	Integer
Default Value	N/A

Min Value	
Max Value	

2.5.4 Tag "interfaceDescription"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/interfaceDescription
Description	This object is an 'alias' name for the interface as specified by a network manager, and provides a non-volatile 'handle' for the interface. CLI: description
Type	String
Default Value	N/A
Min Value	0
Max Value	64

2.5.5 Tag "MACAddress"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/MACAddress
Description	The interface's address at its protocol sub-layer. For example, for an 802.x interface, this object normally contains a MAC address.
Type	MAC Address
Default Value	N/A
Min Value	
Max Value	

2.5.6 Tag "adminState"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/adminState
Description	The desired state of the interface. The testing(3) state indicates that no operational packets can be passed. CLI: shutdown
Type	Integer

Default Value	1
Min Value	1
Max Value	2

Table Enumerator List

up	1
down	2
testing	3

2.5.7 Tag "linkState"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/linkState
Description	The current operational state of the interface.
Type	Integer
Default Value	N/A

Min Value	1
Max Value	8

Table Enumerator List

up	1
down	2
testing	3
unknown	4
dormant	5
notPresent	6
lowerLayerDown	7
suspended	8

2.5.8 Tag "linkStateTrapEnabled"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/link

	StateTrapEnabled
Description	Indicates whether linkUp/linkDown traps should be generated for this interface. By default, this object should have the value enabled(1) for interfaces which do not operate on 'top' of any other interface (as defined in the ifStackTable), and disabled(2) otherwise.
Type	Integer
Default Value	
Min Value	1
Max Value	2

Table Enumerator List

enabled	1
disabled	2

2.5.9 Tag "mediaType"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/mediaType
Description	The transceiver type of this interface.
Type	Integer
Default Value	N/A
Min Value	1
Max Value	4

Table Enumerator List

regular	1
fiberOptics	2
comboRegular	3
comboFiberOptics	4

2.5.10 Tag "adminComboMode"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/adminComboMode
Description	The administrative mode of a combo Ethernet interface. SET: optional if mediaType = 3(comboRegular) or 4(comboFiberOptics), ignored otherwise. CLI: combo mode
Type	Integer
Default Value	
Min Value	1
Max Value	4

Table Enumerator List

force-fiber	1
force-copper	2
prefer-fiber	3
prefer-copper	4

2.5.11 Tag " speedAdmin"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/speedAdmin
Description	his variable specifies the required speed of this interface in bits per second. This specification will take effect only if swIfSpeedDuplexAutoNegotiation is disabled. A value of 10 is returned for 10G. A value of 0 is returned if the value of the variable hasn't been set. CLI: speed
Type	Integer
Default Value	0
Min Value	0
Max Value	10000

2.5.12 Tag " speedOper"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/speedOper

Description	An estimate of the interface's current bandwidth in units of 1,000,000 bits per second. If this object reports a value of 'n' then the speed of the interface is somewhere in the range of 'n-500,000' to 'n+499,999'. For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth.
Type	Integer
Default Value	N/A
Min Value	
Max Value	

2.5.13 Tag "duplexOperMode"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/duplexOperMode
Description	This variable indicates whether this interface operates in half duplex or full duplex mode. This variable can have the values hybrid or unknown only for a trunk. Unknown - only if trunk operative status is not present.

Type	Integer
Default Value	N/A
Min Value	1
Max Value	4

Table Enumerator List

half	1
full	2
hybrid	3
unknown	4

2.5.14 Tag " duplexOperModeDot3"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/duplexOperModeDot3
Description	Not used in CLI. The current mode of operation of the MAC entity. 'unknown'

	indicates that the current duplex mode could not be determined. Management control of the duplex mode is accomplished through the MAU MIB. When an interface does not support autonegotiation, or when autonegotiation is not enabled, the duplex mode is controlled using ifMauDefaultType. When autonegotiation is supported and enabled, duplex mode is controlled using ifMauAutoNegAdvertisedBits. In either case, the currently operating duplex mode is reflected both in this object and in ifMauType. Note that this object provides redundant information with ifMauType. Normally, redundant objects are discouraged. However, in this instance, it allows a management application to determine the duplex status of an interface without having to know every possible value of ifMauType. This was felt to be sufficiently valuable to justify the redundancy.
Type	Integer
Default Value	N/A
Min Value	1
Max Value	3

Table Enumerator List

unknown	1
halfDuplex	2
fullDuplex	3

2.5.15 Tag " duplexAdminMode"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/duplexAdminMode
Description	This variable specifies whether this interface should operate in half duplex or full duplex mode. This specification will take effect only if swIfSpeedDuplexAutoNegotiation is disabled. CLI: duplex
Type	Integer
Default Value	1
Min Value	2
Max Value	3

Table Enumerator List

half	2
full	3

2.5.16 Tag " autoNegotiationOperEnabled"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/autoNegotiationOperEnabled
Description	If enabled the speed and duplex are determined by the device through the autonegotiation process. If disabled these characteristics are determined according to the values of swIfSpeedAdminMode and swIfDuplexAdminMode. hybrid - only for a trunk. unknown - only for ports that there operative status is not present.
Type	Integer
Default Value	N/A
Min Value	1
Max Value	4

Table Enumerator List

enabled	1
disabled	2
hybrid	3
unknown	4

2.5.17 Tag "autoNegotiationAdminEnabled"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/autoNegotiationAdminEnabled
Description	If enabled the speed and duplex mode will be set by the device through the autonegotiation process. Otherwise these characteristics will be set according to the values of swIfSpeedAdminMode and swIfSpeedDuplexAutoNegotiation. CLI: negotiation
Type	TruthValue
Default Value	1
Min Value	1

Max Value	2

2.5.18 Tag "backPressureAdminEnabled"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/backPressureAdminEnabled
Description	This variable indicates whether this interface activates back pressure when congested. CLI: back-pressure
Type	TruthValue
Default Value	2
Min Value	1
Max Value	2

2.5.19 Tag "backPressureOperEnabled"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/backPressureOperEnabled

Description	The operative back pressure mode of this interface.
Type	Integer
Default Value	N/A
Min Value	1
Max Value	4

Table Enumerator List

enabled	1
disabled	2
hybrid	3
unknown	4

2.5.20 Tag "flowControlAdminType"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/flowControlAdminType

Description	CLI: flowcontrol on - Flow control will be enabled on this interface according to the IEEE 802.3x standard. off - Flow control is disabled. autoNegotiation - Flow control will be enabled or disabled on this interface. If enabled, it will operate as specified by the IEEE 802.3x standard. enabledRx - Flow control will be enabled on this interface for recieved frames. enabledTx - Flow control will be enabled on this interface for transmitted frames. An attempt to set this object to 'enabledRx(4)' or 'enabledTx(5)' will fail on interfaces that do not support operation at greater than 100 Mb/s. In any case, flow control can work only if swIfDuplexOperMode is full.
Type	Integer
Default Value	2
Min Value	1
Max Value	5

Table Enumerator List

on	1
off	2

autoNegotiation	3
enabledRx	4
enabledTx	5

2.5.21 Tag "flowControlOperType"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/flowControlOperType
Description	on - Flow control is enabled on this interface according to the IEEE 802.3x standard. off - Flow control is disabled. enabledRx - Flow control is enabled on this interface for recieved frames. enabledTx - Flow control is enabled on this interface for transmitted frames.
Type	Integer
Default Value	N/A
Min Value	1
Max Value	4

Table Enumerator List

on	1
off	2
enabledRx	3
enabledTx	4

2.5.22 Tag " adminAdvertisementList"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/adminAdvertisementList
Description	Administrative auto negotiation capabilities of the interface that can be advertised when swIfSpeedDuplexAutoNegotiation is enabled. default bit means advertise all the port's capabilities according to its type. CLI: negotiation
Type	Bitmap
Default Value	0
Min Value	0

Max Value	7

Table Bitmap List

default	0
unknown	1
tenHalf	2
tenFull	3
fastHalf	4
fastFull	5
gigaHalf	6
gigaFull	7

2.5.23 Tag " operAdvertisementList"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/operAdvertisementList

Description	Operative auto negotiation capabilities of the remote link. unknown bit means that port operative status is not up.
Type	Bitmap
Default Value	N/A
Min Value	1
Max Value	7

Table Bitmap List

default	0
unknown	1
tenHalf	2
tenFull	3
fastHalf	4
fastFull	5
gigaHalf	6

gigaFull	7

2.5.24 Tag "MDIAdminMode"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/MDIAdminMode
Description	The configuration is on a physical port, not include trunks. cross - The interface should force crossover. normal - The interface should not force crossover. auto - Auto mdix is enabled on the interface. CLI: mdix
Type	Integer
Default Value	3
Min Value	1
Max Value	3

Table Enumerator List

cross	1

normal	2
auto	3

2.5.25 Tag "MDIOperMode"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/MDIOperMode
Description	cross - The interface is in crossover mode. normal - The interface is not in crossover mode. unknown - Only for port that its operative status is not present or down.
Type	Integer
Default Value	N/A
Min Value	1
Max Value	3

Table Enumerator List

cross	1

normal	2
unknown	3

2.5.26 Tag "reActivate"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/reActivate
Description	This variable reactivates (enables) an ifIndex (which was suspended). GET value is always 2(false). CLI: set interface active
Type	TruthValue
Default Value	2
Min Value	1
Max Value	2

2.5.27 Tag "LACPEnabled"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSet

	ting/Standard802_3List/Entry/LACPEnabled
Description	The status of LACP_Activity transmitted by the Actor port in LACPDUs.
Type	TruthValue
Default Value	2
Min Value	1
Max Value	2

2.5.28 Tag "LAGID"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/LAGID
Description	The current administrative value of the Key for the Aggregation Port. It is equal to the index of LAG.
Type	Integer
Default Value	0

Min Value	1
Max Value	8

2.5.29 Tag "interfaceName"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/PVEGroupUplink/interfaceName
Description	The unique textual name of the interface.
Type	String
Default Value	N/A
Min Value	
Max Value	

2.5.30 Tag "interfaceID"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/PVEGroupUplink/interfaceID

Description	Obsolete.
Type	Integer
Default Value	
Min Value	
Max Value	

2.5.31 Tag "interfaceType"

Table Tag Attributes

Path	/DeviceConfiguration/InterfaceSetting/Standard802_3List/Entry/PVEGroupUplink/interfaceType
Description	Obsolete.
Type	Integer
Default Value	N/A
Min Value	1
Max Value	2

Table Enumerator List

port	1
LAG	2

2.6 Section "IPv4InterfaceList"

Table Action List

get	Get entry (entries)
set	Add enty (entries); or replace entry; or update specified field(s) of entry (entries)
delete	Delete entry (entries), or restore to default the single entry

Table Section Structure

XML Structure
<IPv4InterfaceList action=" ?" > <ifEntry> <IPAddr/> <interfaceName/> <subnetMask/> <owner/> </ifEntry>

</IPv4InterfaceList>

Table Section Description

Description
CLI: ip address (interface context); show ip interface Query: IPAddr Filter: IPAddr, interfaceName, subnetMask, owner

Table Data Nodes

Tag Name	MIB Name	Access	Key	Mandatory
IPAddr	ipAdEntAddr	R/W	Yes	Yes
interfaceName	ipAdEntIfIndex || rlManagementIfIndex || rlVlanDefaultVID	R/W	No	Yes
subnetMask	ipAdEntNetMask	R/W	No	Yes
owner	rsIpAdEntOwner	R/W	No	Yes

2.6.1 Tag "IPAddr"

Table Tag Attributes

Path	/DeviceConfiguration/IPv4AddressingSetting/IPv4InterfaceList/ifEntry/IPAddr
Description	The IP address to which this entry's addressing information pertains.
Type	IPv4
Default Value	N/A
Min Value	
Max Value	

2.6.2 Tag "interfaceName"

Table Tag Attributes

Path	/DeviceConfiguration/IPv4AddressingSetting/IPv4InterfaceList/ifEntry/interfaceName
Description	The index value which uniquely identifies the interface to which this entry is applicable. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.

	If host parameter management_interface_support=false, the ifIndex should be get from ipAdEntIfIndex. Otherwise, it should be get from rlManagementIfIndex; then, if rlManagementIfIndex=management_interface_default_vlan_ifIndex, it should be calculated from rlVlanDefaultVID.
Type	String
Default Value	N/A
Min Value	
Max Value	

2.6.3 Tag "subnetMask"

Table Tag Attributes

Path	/DeviceConfiguration/IPv4AddressingSetting/IPv4InterfaceList/ifEntry/subnetMask
Description	The subnet mask associated with the IP address of this entry. The value of the mask is an IP address with all the network bits set to 1 and all the hosts bits set to 0.
Type	IPv4 network mask

Default Value	N/A
Min Value	
Max Value	

2.6.4 Tag "owner"

Table Tag Attributes

Path	/DeviceConfiguration/IPv4AddressingSetting/IPv4InterfaceList/ifEntry/owner
Description	The IP Interface owner. Static if interface defined by user, dhcp if received by boot protocol like DHCP and internal for internal usage. Only options 'static', 'dhcp' and 'bootp' may be set by user.
Type	Integer
Default Value	1
Min Value	1
Max Value	4

Table Enumerator List

static	1
dhcp	2
internal	3
default	4
bootp	5