



Cisco Nexus 3000 Series NX-OS Programmability Guide, Release 6.x

First Published: September 15, 2014

Last Modified: July 08, 2016

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883



CONTENTS

Preface

Preface v

Audience v

Document Conventions v

Related Documentation for Cisco Nexus 3000 Series NX-OS Software vii

Documentation Feedback viii

Obtaining Documentation and Submitting a Service Request viii

CHAPTER 1

Bash 1

About Bash 1

Accessing Bash 1

Escalate Privileges to Root 2

Examples of Bash Commands 3

 Displaying System Statistics 3

 Running Bash from CLI 3

 Running Python from Bash 4

CHAPTER 2

NX-API 5

About NX-API 5

 Feature NX-API 5

 Transport 6

 Message Format 6

 Security 6

Using NX-API 7

 NX-API Sandbox 9

 NX-API Management Commands 10

 NX-API Request Elements 11

 NX-API Response Elements 14

About JSON (JavaScript Object Notation) 15
CLI Execution 15
XML and JSON Supported Commands 16
Examples of XML and JSON Output 17

CHAPTER 3

NX-API Response Codes 19

Table of NX-API Response Codes 19

CHAPTER 4

XML Support for ABM and LM in N3500 21

XML Support for ABM and LM in N3500 21



Preface

The preface contains the following sections:

- [Audience, page v](#)
- [Document Conventions, page v](#)
- [Related Documentation for Cisco Nexus 3000 Series NX-OS Software, page vii](#)
- [Documentation Feedback, page viii](#)
- [Obtaining Documentation and Submitting a Service Request, page viii](#)

Audience

This publication is for network administrators who configure and maintain Cisco Nexus devices.

Document Conventions



Note

As part of our constant endeavor to remodel our documents to meet our customers' requirements, we have modified the manner in which we document configuration tasks. As a result of this, you may find a deviation in the style used to describe these tasks, with the newly included sections of the document following the new format.

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
<i>Italic</i>	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element (keyword or argument).

Convention	Description
[x y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.
[x {y z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.
<i>variable</i>	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Examples use the following conventions:

Convention	Description
<code>screen font</code>	Terminal sessions and information the switch displays are in screen font.
<code>boldface screen font</code>	Information you must enter is in boldface screen font.
<i><code>italic screen font</code></i>	Arguments for which you supply values are in italic screen font.
<>	Nonprinting characters, such as passwords, are in angle brackets.
[]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

This document uses the following conventions:



Note

Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.



Caution

Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Related Documentation for Cisco Nexus 3000 Series NX-OS Software

The entire Cisco NX-OS 3000 Series documentation set is available at the following URL:

http://www.cisco.com/en/US/products/ps11541/tsd_products_support_series_home.html

Release Notes

The release notes are available at the following URL:

http://www.cisco.com/en/US/products/ps11541/prod_release_notes_list.html

Installation and Upgrade Guides

The installation and upgrade guides are available at the following URL:

http://www.cisco.com/en/US/products/ps11541/prod_installation_guides_list.html

License Information

For information about feature licenses in NX-OS, see the *Cisco NX-OS Licensing Guide*, available at the following URL: http://www.cisco.com/en/US/docs/switches/datacenter/sw/nx-os/licensing/guide/b_Cisco_NX-OS_Licensing_Guide.html.

For the NX-OS end user agreement and copyright information, see *License and Copyright Information for Cisco NX-OS Software*, available at the following URL: http://www.cisco.com/en/US/docs/switches/datacenter/sw/4_0/nx-os/license_agreement/nx-ossw_lisns.html.

Configuration Guides

The configuration guides are available at the following URL:

http://www.cisco.com/en/US/products/ps11541/products_installation_and_configuration_guides_list.html

Programming Guides

The XML Interface User Guide and other programming guides are available at the following URL:

http://www.cisco.com/en/US/products/ps11541/products_programming_reference_guides_list.html

Technical References

The technical references are available at the following URL:

http://www.cisco.com/en/US/products/ps11541/prod_technical_reference_list.html

Error and System Messages

The error and system message reference guides are available at the following URL:

http://www.cisco.com/en/US/products/ps11541/products_system_message_guides_list.html

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to: ciscodfa-docfeedback@cisco.com.

We appreciate your feedback.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see [What's New in Cisco Product Documentation](#).

To receive new and revised Cisco technical content directly to your desktop, you can subscribe to the [What's New in Cisco Product Documentation RSS feed](#). RSS feeds are a free service.



CHAPTER

1

Bash

- [About Bash, page 1](#)
- [Accessing Bash, page 1](#)
- [Escalate Privileges to Root, page 2](#)
- [Examples of Bash Commands, page 3](#)

About Bash

In addition to the NX-OS CLI, Cisco Nexus 3000 Series devices support access to the Bourne-Again SHell (Bash). Bash interprets commands that you enter or commands that are read from a shell script. Using Bash enables access to the underlying Linux system on the device and to manage the system.

Accessing Bash

In Cisco NX-OS, Bash is accessible from user accounts that are associated with the Cisco NX-OS dev-ops role or the Cisco NX-OS network-admin role.

The following example shows the authority of the dev-ops role and the network-admin role:

```
switch# show role name dev-ops

Role: dev-ops
Description: Predefined system role for devops access. This role
cannot be modified.
Vlan policy: permit (default)
Interface policy: permit (default)
Vrf policy: permit (default)
-----
Rule      Perm      Type      Scope      Entity
-----
4         permit   command   conf t ;   username *
3         permit   command   bcm module *
2         permit   command   run bash *
1         permit   command   python *
switch# show role name network-admin

Role: network-admin
Description: Predefined network admin role has access to all commands
on the switch
-----
```

```

Rule      Perm      Type      Scope      Entity
-----
1         permit   read-write
switch#

```

Bash is enabled by running the **feature bash-shell** command.

The **run bash** command loads Bash and begins at the home directory for the user.

The following examples show how to enable the Bash shell feature and how to run Bash.

```

switch# configure terminal
switch(config)# feature bash-shell
switch# run bash
Linux# whoami
admin
Linux# pwd
/bootflash/home/admin
Linux#

```

**Note**

You can also execute Bash commands with the **run bash <command>** command.

The following is an example of the **run bash <command>** command.

```
run bash whoami
```

Escalate Privileges to Root

The privileges of an admin user can escalate their privileges for root access.

The following are guidelines for escalating privileges:

- Only an admin user can escalate privileges to root.
- Bash must be enabled before escalating privileges.
- Escalation to root is password protected.

The following example shows how to escalate privileges to root and how to verify the escalation:

```

switch# run bash
Linux# sudo su root

```

We trust you have received the usual lecture from the local System Administrator. It usually boils down to these three things:

- ```

#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

```

Password:

```

Linux# whoami
root
Linux# exit
exit

```

# Examples of Bash Commands

This section contains examples of Bash commands and output.

## Displaying System Statistics

The following example shows how to display system statistics:

```
switch# run bash
Linux# cat /proc/meminfo
MemTotal: 3795100 kB
MemFree: 1472680 kB
Buffers: 136 kB
Cached: 1100116 kB
ShmFS: 1100116 kB
Allowed: 948775 Pages
Free: 368170 Pages
Available: 371677 Pages
SwapCached: 0 kB
Active: 1198872 kB
Inactive: 789764 kB
SwapTotal: 0 kB
SwapFree: 0 kB
Dirty: 0 kB
Writeback: 0 kB
AnonPages: 888272 kB
Mapped: 144044 kB
Slab: 148836 kB
SReclaimable: 13892 kB
SUnreclaim: 134944 kB
PageTables: 28724 kB
NFS_Unstable: 0 kB
Bounce: 0 kB
WritebackTmp: 0 kB
CommitLimit: 1897548 kB
Committed_AS: 19984932 kB
VmallocTotal: 34359738367 kB
VmallocUsed: 215620 kB
VmallocChunk: 34359522555 kB
HugePages_Total: 0
HugePages_Free: 0
HugePages_Rsvd: 0
HugePages_Surp: 0
Hugepagesize: 2048 kB
DirectMap4k: 40960 kB
DirectMap2M: 4190208 kB
Linux#
```

## Running Bash from CLI

The following example shows how to run a bash command from the CLI with the `run bash <command>` command:

```
switch# run bash ps -el
F S UID PID PPID C PRI NI ADDR SZ WCHAN TTY TIME CMD
4 S 0 1 0 0 80 0 - 497 select ? 00:00:08 init
5 S 0 2 0 0 75 -5 - 0 kthrea ? 00:00:00 kthreadd
1 S 0 3 2 0 -40 - - 0 migrat ? 00:00:00 migration/0
1 S 0 4 2 0 75 -5 - 0 ksofti ? 00:00:01 ksoftirqd/0
5 S 0 5 2 0 58 - - 0 watchd ? 00:00:00 watchdog/0
1 S 0 6 2 0 -40 - - 0 migrat ? 00:00:00 migration/1
1 S 0 7 2 0 75 -5 - 0 ksofti ? 00:00:00 ksoftirqd/1
5 S 0 8 2 0 58 - - 0 watchd ? 00:00:00 watchdog/1
```

```

1 S 0 9 2 0 -40 - - 0 migrat ? 00:00:00 migration/2
1 S 0 10 2 0 75 -5 - 0 ksofti ? 00:00:00 ksoftirqd/2
5 S 0 11 2 0 58 - - 0 watchd ? 00:00:00 watchdog/2
1 S 0 12 2 0 -40 - - 0 migrat ? 00:00:00 migration/3
1 S 0 13 2 0 75 -5 - 0 ksofti ? 00:00:00 ksoftirqd/3
5 S 0 14 2 0 58 - - 0 watchd ? 00:00:00 watchdog/3

...

4 S 0 8864 1 0 80 0 - 2249 wait ttyS0 00:00:00 login
4 S 2002 28073 8864 0 80 0 - 69158 select ttyS0 00:00:00 vsh
4 R 0 28264 3782 0 80 0 - 54790 select ? 00:00:00 in.dcos-telnet
4 S 0 28265 28264 0 80 0 - 2247 wait pts/0 00:00:00 login
4 S 2002 28266 28265 0 80 0 - 69175 wait pts/0 00:00:00 vsh
1 S 2002 28413 28266 0 80 0 - 69175 wait pts/0 00:00:00 vsh
0 R 2002 28414 28413 0 80 0 - 887 - pts/0 00:00:00 ps
switch#

```

## Running Python from Bash

The following example shows how to load Python and configure a switch using Python objects:

```

switch# run bash
Linux# python
Python 2.7.5 (default, May 16 2014, 10:58:01)
[GCC 4.3.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Loaded cisco NxOS lib!
>>>
>>> from cisco import *
>>> from cisco.vrf import *
>>> from cisco.interface import *
>>> vrfobj=VRF('myvrf')
>>> vrfobj.get_name()
'myvrf'
>>> vrfobj.add_interface('Ethernet1/3')
True
>>> intf=Interface('Ethernet1/3')
>>> print intf.config()

!Command: show running-config interface Ethernet1/3
!Time: Thu Aug 21 23:32:25 2014

version 6.0(2)U4(1)

interface Ethernet1/3
 no switchport
 vrf member myvrf

>>>

```



## NX-API

---

- [About NX-API, page 5](#)
- [Using NX-API, page 7](#)
- [XML and JSON Supported Commands, page 16](#)

### About NX-API

On Cisco Nexus devices, command-line interfaces (CLIs) are run only on the device. NX-API improves the accessibility of these CLIs by making them available outside of the switch by using HTTP/HTTPS. You can use this extension to the existing Cisco Nexus CLI system on the Cisco Nexus 3000 Series devices. NX-API supports **show** commands, configurations, and Linux Bash.

NX-API supports JSON-RPC.

### Feature NX-API

- Feature NX-API is required to be enabled for access the device through sandbox.
- `| json` on the device internally uses python script to generate output.
- NX-API can be enabled either on http/https via ipv4 or ipv6

```
BLR-VXLAN-NPT-CR-179# show nxapi
nxapi enabled
HTTP Listen on port 80
HTTPS Listen on port 443
BLR-VXLAN-NPT-CR-179#
```
- NX-API is internally spawning third-party NGINX process, which handler receive/send/processing of http requests/response

```
nxapi certificate {httpsCRT |httpskey}
nxapi certificate enable
```
- NX-API Certificates can be enabled for https
- Default port for nginx to operate is 80/443 for http/https respectively. It can also be changed using the following CLI's

```
nxapi {http|https} port port
```

## Transport

NX-API uses HTTP/HTTPS as its transport. CLIs are encoded into the HTTP/HTTPS POST body.

The NX-API backend uses the Nginx HTTP server. The Nginx process, and all of its children processes, are under Linux cgroup protection where the CPU and memory usage is capped. If the Nginx memory usage exceeds the cgroup limitations, the Nginx process is restarted and restored.

## Message Format

**Note**

- NX-API XML output presents information in a user-friendly format.
- NX-API XML does not map directly to the Cisco NX-OS NETCONF implementation.
- NX-API XML output can be converted into JSON.

## Security

NX-API supports HTTPS. All communication to the device is encrypted when you use HTTPS.

NX-API is integrated into the authentication system on the device. Users must have appropriate accounts to access the device through NX-API. NX-API uses HTTP basic authentication. All requests must contain the username and password in the HTTP header.

**Note**

You should consider using HTTPS to secure your user's login credentials.

You can enable NX-API by using the **feature** manager CLI command. NX-API is disabled by default.

NX-API provides a session-based cookie, **nxapi\_auth** when users first successfully authenticate. With the session cookie, the username and password are included in all subsequent NX-API requests that are sent to the device. The username and password are used with the session cookie to bypass performing the full authentication process again. If the session cookie is not included with subsequent requests, another session cookie is required and is provided by the authentication process. Avoiding unnecessary use of the authentication process helps to reduce the workload on the device.

**Note**

A **nxapi\_auth** cookie expires in 600 seconds (10 minutes). This value is a fixed and cannot be adjusted.

**Note**

NX-API performs authentication through a programmable authentication module (PAM) on the switch. Use cookies to reduce the number of PAM authentications, which reduces the load on the PAM.

## Using NX-API

The commands, command type, and output type for the Cisco Nexus 3000 Series devices are entered using NX-API by encoding the CLIs into the body of a HTTP/HTTPS POST. The response to the request is returned in XML or JSON output format.



**Note** For more details about NX-API response codes, see [Table of NX-API Response Codes](#), on page 19.

You must enable NX-API with the **feature** manager CLI command on the device. By default, NX-API is disabled.

The following example shows how to configure and launch the NX-API Sandbox:

- Enable the management interface.

```
switch# conf t
switch(config)# interface mgmt 0
switch(config)# ip address 198.51.100.1/24
switch(config)# vrf context management
switch(config)# ip route 203.0.113.1/0 1.2.3.1
```

- Enable the NX-API **nxapi** feature.

```
switch# conf t
switch(config)# feature nxapi
```

The following example shows a request and its response in XML format:

Request:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ins_api>
 <version>0.1</version>
 <type>cli_show</type>
 <chunk>0</chunk>
 <sid>session1</sid>
 <input>show switchname</input>
 <output_format>xml</output_format>
</ins_api>
```

Response:

```
<?xml version="1.0"?>
<ins_api>
 <type>cli_show</type>
 <version>0.1</version>
 <sid>eoc</sid>
 <outputs>
 <output>
 <body>
 <hostname>switch</hostname>
 </body>
 <input>show switchname</input>
 <msg>Success</msg>
 <code>200</code>
 </output>
 </outputs>
</ins_api>
```

The following example shows a request and its response in JSON format:

**Request:**

```
{
 "ins_api": {
 "version": "0.1",
 "type": "cli_show",
 "chunk": "0",
 "sid": "session1",
 "input": "show switchname",
 "output_format": "json"
 }
}
```

**Response:**

```
{
 "ins_api": {
 "type": "cli_show",
 "version": "0.1",
 "sid": "eoc",
 "outputs": {
 "output": {
 "body": {
 "hostname": "switch"
 },
 "input": "show switchname",
 "msg": "Success",
 "code": "200"
 }
 }
 }
}
```

**Using the Management Interface for NX-API calls**

It is recommended to use the management interface for NX-API calls.

When using non-management interface and a custom port for NX-API an entry should be made in the CoPP policy to prevent NX-API traffic from hitting the default copp entry which could unfavorably treat API traffic.

**Note**

It is recommended to use the management interface for NX-API traffic. If that is not possible and a custom port is used, the "copp-http" class should be updated to include the custom NX-API port.

The following example port 9443 is being used for NX-API traffic.

This port is added to the copp-system-acl-http ACL to allow it to be matched under the copp-http class resulting on 100 pps policing. (This may need to be increased in certain environments.)

```
!
ip access-list copp-system-acl-http
 10 permit tcp any any eq www
 20 permit tcp any any eq 443
 30 permit tcp any any eq 9443 <-----
!
class-map type control-plane match-any copp-http
 match access-group name copp-system-acl-http
!
policy-map type control-plane copp-system-policy
 class copp-http
 police pps 100
!
```



## NX-API Sandbox

The NX-API Sandbox is the web-based user interface that you use to enter the commands, command type, and output type for the Cisco Nexus 3000 Series device using HTTP/HTTPS. After posting the request, the output response is displayed.

By default, NX-API is disabled. Begin enabling NX-API with the **feature** manager CLI command on the switch. Then enable NX-API with the **nxapi sandbox** command.

Use a browser to access the NX-API Sandbox.

**Note**

---

When using the NX-API Sandbox, Cisco recommends that you use the Firefox browser, release 24.0 or later.

---

You can also NX-API over IPv6.

The following example shows how to configure and launch the NX-API Sandbox:

- Enable the management interface.

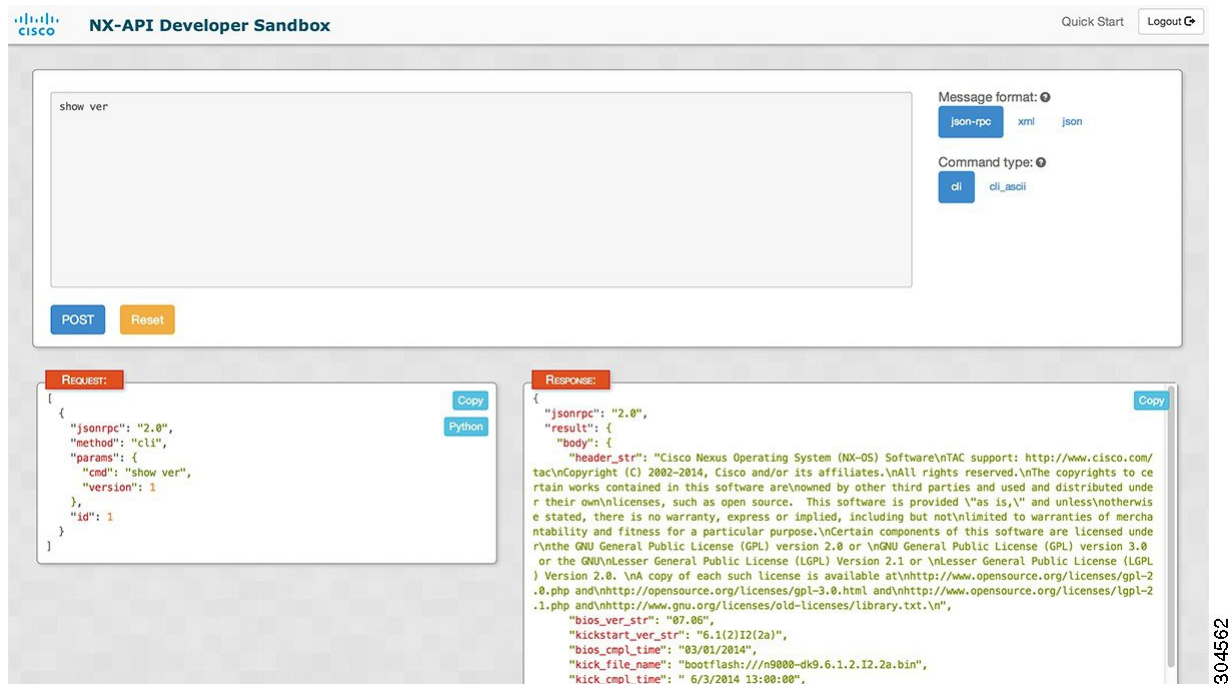
```
switch# conf t
switch(config)# interface mgmt 0
switch(config)# ip address 198.51.100.1/24
switch(config)# vrf context management
switch(config)# ip route 203.0.113.1/0 1.2.3.1
```

- Enable the NX-API **nxapi** feature.

```
switch# conf t
switch(config)# feature nxapi
switch(config)# nxapi sandbox
```

- Open a browser and enter `http://mgmt-ip` to launch the NX-API Sandbox. The following figure is an example of a request and output response.

**Figure 1: NX-API Sandbox with Example Request and Output Response**



In the NX-API Sandbox, you specify the commands, command type, and output type in the top pane. Click the POST Request button above the left pane to post the request. Brief descriptions of the request elements are displayed below the left pane.

After the request is posted, the output response is displayed in the right pane.

The following sections describe the commands to manage NX-API and descriptions of the elements of the request and the output response.

## NX-API Management Commands

You can enable and manage NX-API with the CLI commands listed in the following table.

**Table 1: NX-API Management Commands**

| NX-API Management Command                 | Description       |
|-------------------------------------------|-------------------|
| <code>feature nxapi</code>                | Enables NX-API.   |
| <code>no feature nxapi</code>             | Disables NX-API.  |
| <code>nxapi {http https} port port</code> | Specifies a port. |

| NX-API Management Command                                      | Description                                                                                                                                                                                             |
|----------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>no nxapi</b> {http https}                                   | Disables HTTP/HTTPS.                                                                                                                                                                                    |
| <b>show nxapi</b>                                              | Displays port information.                                                                                                                                                                              |
| <b>nxapi certificate</b> { <i>httpsert</i>   <i>httpskey</i> } | Specifies the upload of the following: <ul style="list-style-type: none"> <li>• HTTPS certificate when <i>httpsert</i> is specified.</li> <li>• HTTPS key when <i>httpskey</i> is specified.</li> </ul> |
| <b>nxapi certificate</b> <i>enable</i>                         | Enables a certificate.                                                                                                                                                                                  |

## NX-API Request Elements

NX-API request elements are sent to the device in XML format or JSON format. The HTTP header of the request must identify the content type of the request.

You use the NX-API elements that are listed in the following table to specify a CLI command:

**Table 2: NX-API Request Elements**

| NX-API Request Element | Description                   |
|------------------------|-------------------------------|
| version                | Specifies the NX-API version. |

| NX-API Request Element | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>type</i>            | <p>Specifies the type of command to be executed.</p> <p>The following types of commands are supported:</p> <ul style="list-style-type: none"> <li>• <b>cli_show</b><br/>CLI <b>show</b> commands that expect structured output. If the command does not support XML output, an error message is returned.</li> <li>• <b>cli_show_ascii</b><br/>CLI <b>show</b> commands that expect ASCII output. This aligns with existing scripts that parse ASCII output. Users are able to use existing scripts with minimal changes.</li> <li>• <b>cli_conf</b><br/>CLI configuration commands.</li> <li>• <b>bash</b><br/>Bash commands. Most non-interactive Bash commands are supported by NX-API.</li> </ul> <p><b>Note</b></p> <ul style="list-style-type: none"> <li>• Each command is only executable with the current user's authority.</li> <li>• The pipe operation is supported in the output when the message type is ASCII. If the output is in XML format, the pipe operation is not supported.</li> <li>• A maximum of 10 consecutive <b>show</b> commands are supported. If the number of <b>show</b> commands exceeds 10, the 11th and subsequent commands are ignored.</li> <li>• No interactive commands are supported.</li> </ul> |

| NX-API Request Element | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |          |                                                 |          |                                         |      |                             |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|-------------------------------------------------|----------|-----------------------------------------|------|-----------------------------|
| <p><i>chunk</i></p>    | <p>Some <b>show</b> commands can return a large amount of output. For the NX-API client to start processing the output before the entire command completes, NX-API supports output chunking for <b>show</b> commands.</p> <p>Enable or disable chunk with the following settings:</p> <table border="1" data-bbox="824 464 1515 590"> <tr> <td data-bbox="824 464 938 527">0</td> <td data-bbox="938 464 1515 527">Do not chunk output.</td> </tr> <tr> <td data-bbox="824 527 938 590">1</td> <td data-bbox="938 527 1515 590">Chunk output.</td> </tr> </table> <p><b>Note</b> Only <b>show</b> commands support chunking. When a series of <b>show</b> commands are entered, only the first command is chunked and returned.</p> <p>The output message format is XML. (XML is the default.) Special characters, such as &lt; or &gt;, are converted to form a valid XML message (&lt; is converted into &amp;lt; &gt; is converted into &amp;gt;).</p> <p><b>Note</b> You can use XML SAX to parse the chunked output. When chunking is enabled, the message format is limited to XML. JSON output format is not supported when chunking is enabled.</p> | 0        | Do not chunk output.                            | 1        | Chunk output.                           |      |                             |
| 0                      | Do not chunk output.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |          |                                                 |          |                                         |      |                             |
| 1                      | Chunk output.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |          |                                                 |          |                                         |      |                             |
| <p><i>sid</i></p>      | <p>The session ID element is valid only when the response message is chunked. To retrieve the next chunk of the message, you must specify a <i>sid</i> to match the <i>sid</i> of the previous response message.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |          |                                                 |          |                                         |      |                             |
| <p><i>input</i></p>    | <p>Input can be one command or multiple commands. However, commands that belong to different message types should not be mixed. For example, <b>show</b> commands are cli_show message type and are not supported in cli_conf mode.</p> <p><b>Note</b> Except for <b>bash</b>, multiple commands are separated with " ; ". (The ; must be surrounded with single blank characters.)</p> <p>For <b>bash</b>, multiple commands are separated with ";". (The ; is <b>not</b> surrounded with single blank characters.)</p> <p>The following are examples of multiple commands:</p> <table border="1" data-bbox="824 1520 1515 1751"> <tr> <td data-bbox="824 1520 948 1598">cli_show</td> <td data-bbox="948 1520 1515 1598">show version ; show interface brief ; show vlan</td> </tr> <tr> <td data-bbox="824 1598 948 1675">cli_conf</td> <td data-bbox="948 1598 1515 1675">interface Eth4/1 ; no shut ; switchport</td> </tr> <tr> <td data-bbox="824 1675 948 1751">bash</td> <td data-bbox="948 1675 1515 1751">cd /bootflash;mkdir new_dir</td> </tr> </table>                                                                                        | cli_show | show version ; show interface brief ; show vlan | cli_conf | interface Eth4/1 ; no shut ; switchport | bash | cd /bootflash;mkdir new_dir |
| cli_show               | show version ; show interface brief ; show vlan                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |          |                                                 |          |                                         |      |                             |
| cli_conf               | interface Eth4/1 ; no shut ; switchport                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |          |                                                 |          |                                         |      |                             |
| bash                   | cd /bootflash;mkdir new_dir                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |          |                                                 |          |                                         |      |                             |

| NX-API Request Element | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |     |                                 |      |                                  |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|---------------------------------|------|----------------------------------|
| <i>output_format</i>   | <p>The available output message formats are the following:</p> <table border="1" data-bbox="789 317 1479 449"> <tbody> <tr> <td data-bbox="789 317 1024 380">xml</td> <td data-bbox="1024 317 1479 380">Specifies output in XML format.</td> </tr> <tr> <td data-bbox="789 380 1024 449">json</td> <td data-bbox="1024 380 1479 449">Specifies output in JSON format.</td> </tr> </tbody> </table> <p><b>Note</b> The Cisco Nexus 3000 Series CLI supports XML output, which means that the JSON output is converted from XML. The conversion is processed on the switch.</p> <p>To manage the computational overhead, the JSON output is determined by the amount of output. If the output exceeds 1 MB, the output is returned in XML format. When the output is chunked, only XML output is supported.</p> <p>The content-type header in the HTTP/HTTPS headers indicate the type of response format (XML or JSON).</p> | xml | Specifies output in XML format. | json | Specifies output in JSON format. |
| xml                    | Specifies output in XML format.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |     |                                 |      |                                  |
| json                   | Specifies output in JSON format.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |     |                                 |      |                                  |

## NX-API Response Elements

The NX-API elements that respond to a CLI command are listed in the following table:

**Table 3: NX-API Response Elements**

| NX-API Response Element | Description                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| version                 | NX-API version.                                                                                                                                                                                                                                                                                                                                                                                              |
| type                    | Type of command to be executed.                                                                                                                                                                                                                                                                                                                                                                              |
| sid                     | Session ID of the response. This element is valid only when the response message is chunked.                                                                                                                                                                                                                                                                                                                 |
| outputs                 | <p>Tag that encloses all command outputs.</p> <p>When multiple commands are in <code>cli_show</code> or <code>cli_show_ascii</code>, each command output is enclosed by a single output tag.</p> <p>When the message type is <code>cli_conf</code> or <code>bash</code>, there is a single output tag for all the commands because <code>cli_conf</code> and <code>bash</code> commands require context.</p> |
| output                  | <p>Tag that encloses the output of a single command output.</p> <p>For <code>cli_conf</code> and <code>bash</code> message types, this element contains the outputs of all the commands.</p>                                                                                                                                                                                                                 |

| NX-API Response Element | Description                                                                                                                                                                                                                                                                                                                                     |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| input                   | Tag that encloses a single command that was specified in the request. This element helps associate a request input element with the appropriate response output element.                                                                                                                                                                        |
| body                    | Body of the command response.                                                                                                                                                                                                                                                                                                                   |
| code                    | Error code returned from the command execution.<br><br>NX-API uses standard HTTP error codes as described by the Hypertext Transfer Protocol (HTTP) Status Code Registry ( <a href="http://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml">http://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml</a> ). |
| msg                     | Error message associated with the returned error code.                                                                                                                                                                                                                                                                                          |

## About JSON (JavaScript Object Notation)

JSON is a light-weight text-based open standard designed for human-readable data and is an alternative to XML. JSON was originally designed from JavaScript, but it is language-independent data format. The JSON/CLI Execution is currently supported in Nexus 3500.



### Note

The NX-API/JSON functionality is now available on the Cisco Nexus 3500 Series platform.

The two primary Data Structures that are supported in some way by nearly all modern programming languages are as follows:

- Ordered List :: Array
- Unordered List (Name/Value pair) :: Objects

JSON /XML output for a show command can also be accessed via sandbox.

## CLI Execution

### Show\_Command | json

#### Example Code

```
BLR-VXLAN-NPT-CR-179# show cdp neighbors | json
{"TABLE_cdp_neighbor_brief_info": {"ROW_cdp_neighbor_brief_info": [{"ifindex": "83886080", "device_id": "SW-SPARSHA-SAVBU-F10", "intf_id": "mgmt0", "ttl": "148", "capability": ["switch", "IGMP_cnd_filtering"], "platform_id": "cisco WS-C2960S-48TS-L", "port_id": "GigabitEthernet1/0/24"}, {"ifindex": "436207616", "device_id": "BLR-VXLAN-NPT-CR-178 (FOC1745R01W)", "intf_id": "Ethernet1/1", "ttl": "166", "capability": ["router", "switch", "IGMP_cnd_filtering", "Supports-STP-Dispute"], "platform_id": "N3K-C3132Q-40G", "port_id": "Ethernet1/1"}]}}
```

## XML and JSON Supported Commands

Cisco NX-OS Release 6.0(2)U4(1) introduces support for XML and JSON output of the following commands:

- show bgp all
- show bgp process
- show bgp convergence
- show bgp ip unicast/multicast
- show bgp ipv4 unicast/multicast
- show bgp ipv6 unicast/multicast
- show bgp paths
- show bgp peer-policy
- show bgp vrf
- show bgp sessions
- show bgp statistics
- show consistency-checker forwarding ipv4
- show consistency-checker forwarding ipv6
- show lldp neighbors
- show lldp neighbors detail
- show lldp neighbors interface ethernet x/x
- show lldp neighbors interface ethernet x/x detail
- show lldp portid-subtype
- show lldp timers
- show lldp tlv-select
- show lldp traffic
- show lldp traffic interface ethernet x/x
- show process memory
- show process cpu & show process
- show routing vrf all
- show system internal forwarding route summary
- show system resources



## Examples of XML and JSON Output

This example shows how to display the unicast and multicast routing entries in hardware tables in JSON format:

```
switch(config)# show hardware profile status | json
{"total_lpm": ["8191", "1024"], "total_host": "8192", "max_host4_limit": "4096",
 "max_host6_limit": "2048", "max_mcast_limit": "2048", "used_lpm_total": "9", "u
sed_v4_lpm": "6", "used_v6_lpm": "3", "used_v6_lpm_128": "1", "used_host_lpm_tot
al": "0", "used_host_v4_lpm": "0", "used_host_v6_lpm": "0", "used_mcast": "0", "
used_mcast_oif1": "2", "used_host_in_host_total": "13", "used_host4_in_host": "1
2", "used_host6_in_host": "1", "max_ecmp_table_limit": "64", "used_ecmp_table":
"0", "mfib_fd_status": "Disabled", "mfib_fd_maxroute": "0", "mfib_fd_count": "0"
}
switch(config)#
```

This example shows how to display the unicast and multicast routing entries in hardware tables in XML format:

```
switch(config)# show hardware profile status | xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://w
ww.cisco.com/nxos:1.0:fib">
 <nf:data>
 <show>
 <hardware>
 <profile>
 <status>
 <_XML_OPT_Cmd_dynamic_tcama_status__readonly__>
 <_readonly__>
 <total_lpm>8191</total_lpm>
 <total_host>8192</total_host>
 <total_lpm>1024</total_lpm>
 <max_host4_limit>4096</max_host4_limit>
 <max_host6_limit>2048</max_host6_limit>
 <max_mcast_limit>2048</max_mcast_limit>
 <used_lpm_total>9</used_lpm_total>
 <used_v4_lpm>6</used_v4_lpm>
 <used_v6_lpm>3</used_v6_lpm>
 <used_v6_lpm_128>1</used_v6_lpm_128>
 <used_host_lpm_total>0</used_host_lpm_total>
 <used_host_v4_lpm>0</used_host_v4_lpm>
 <used_host_v6_lpm>0</used_host_v6_lpm>
 <used_mcast>0</used_mcast>
 <used_mcast_oif1>2</used_mcast_oif1>
 <used_host_in_host_total>13</used_host_in_host_total>
 <used_host4_in_host>12</used_host4_in_host>
 <used_host6_in_host>1</used_host6_in_host>
 <max_ecmp_table_limit>64</max_ecmp_table_limit>
 <used_ecmp_table>0</used_ecmp_table>
 <mfib_fd_status>Disabled</mfib_fd_status>
 <mfib_fd_maxroute>0</mfib_fd_maxroute>
 <mfib_fd_count>0</mfib_fd_count>
 </_readonly__>
 </_XML_OPT_Cmd_dynamic_tcama_status__readonly__>
 </status>
 </profile>
 </hardware>
 </show>
 </nf:data>
</nf:rpc-reply>
]]>]]>
switch(config)#
```

This example shows how to display LLDP timers configured on the switch in JSON format:

```
switch(config)# show lldp timers | json
{"ttl": "120", "reinit": "2", "tx_interval": "30", "tx_delay": "2", "hold_mplier": "4", "notification_interval": "5"}
switch(config)#
```

This example shows how to display LLDP timers configured on the switch in XML format:

```
switch(config)# show lldp timers | xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://www.cisco.com/nxos:1.0:lldp">
 <nf:data>
 <show>
 <lldp>
 <timers>
 <__XML__OPT_Cmd_lldp_show_timers__readonly__>
 <__readonly__>
 <ttl>120</ttl>
 <reinit>2</reinit>
 <tx_interval>30</tx_interval>
 <tx_delay>2</tx_delay>
 <hold_mplier>4</hold_mplier>
 <notification_interval>5</notification_interval>
 </__readonly__>
 </__XML__OPT_Cmd_lldp_show_timers__readonly__>
 </timers>
 </lldp>
 </show>
 </nf:data>
</nf:rpc-reply>
]]]]>
switch(config)#
```



## NX-API Response Codes

- [Table of NX-API Response Codes, page 19](#)

### Table of NX-API Response Codes

The following are the possible NX-API errors, error codes, and messages of an NX-API response.

The following are the possible NX-API errors, error codes, and messages of an NX-API response.



**Note**

The standard HTTP error codes are at the Hypertext Transfer Protocol (HTTP) Status Code Registry (<http://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml>).

**Table 4: NX-API Response Codes**

| NX-API Response         | Code | Message                                   |
|-------------------------|------|-------------------------------------------|
| SUCCESS                 | 200  | Success.                                  |
| CUST_OUTPUT_PIPED       | 204  | Output is piped elsewhere due to request. |
| BASH_CMD_ERR            | 400  | Input Bash command error.                 |
| CHUNK_ALLOW_ONE_CMD_ERR | 400  | Chunking only allowed to one command.     |
| CLI_CLIENT_ERR          | 400  | CLI execution error.                      |
| CLI_CMD_ERR             | 400  | Input CLI command error.                  |
| IN_MSG_ERR              | 400  | Request message is invalid.               |
| NO_INPUT_CMD_ERR        | 400  | No input command.                         |
| PERM_DENY_ERR           | 401  | Permission denied.                        |

|                               |     |                                                                          |
|-------------------------------|-----|--------------------------------------------------------------------------|
| CONF_NOT_ALLOW_SHOW_ERR       | 405 | Configuration mode does not allow <b>show</b> .                          |
| SHOW_NOT_ALLOW_CONF_ERR       | 405 | Show mode does not allow configuration.                                  |
| EXCEED_MAX_SHOW_ERR           | 413 | Maximum number of consecutive show commands exceeded. The maximum is 10. |
| MSG_SIZE_LARGE_ERR            | 413 | Response size too large.                                                 |
| BACKEND_ERR                   | 500 | Backend processing error.                                                |
| FILE_OPER_ERR                 | 500 | System internal file operation error.                                    |
| LIBXML_NS_ERR                 | 500 | System internal LIBXML NS error.                                         |
| LIBXML_PARSE_ERR              | 500 | System internal LIBXML parse error.                                      |
| LIBXML_PATH_CTX_ERR           | 500 | System internal LIBXML path context error.                               |
| MEM_ALLOC_ERR                 | 500 | System internal memory allocation error.                                 |
| USER_NOT_FOUND_ERR            | 500 | User not found from input or cache.                                      |
| XML_TO_JSON_CONVERT_ERR       | 500 | XML to JSON conversion error.                                            |
| BASH_CMD_NOT_SUPPORTED_ERR    | 501 | Bash command not supported.                                              |
| CHUNK_ALLOW_XML_ONLY_ERR      | 501 | Chunking allows only XML output.                                         |
| JSON_NOT_SUPPORTED_ERR        | 501 | JSON not supported due to large amount of output.                        |
| MSG_TYPE_UNSUPPORTED_ERR      | 501 | Message type not supported.                                              |
| PIPE_OUTPUT_NOT_SUPPORTED_ERR | 501 | Pipe operation not supported.                                            |
| PIPE_XML_NOT_ALLOWED_IN_INPUT | 501 | Pipe XML is not allowed in input.                                        |
| RESP_BIG_JSON_NOT_ALLOWED_ERR | 501 | Response has large amount of output. JSON not supported.                 |
| STRUCT_NOT_SUPPORTED_ERR      | 501 | Structured output unsupported.                                           |
| ERR_UNDEFINED                 | 600 | Undefined.                                                               |



## CHAPTER

# 4

## XML Support for ABM and LM in N3500

This chapter contains XML and CLI supported in N3500.

- [XML Support for ABM and LM in N3500](#) , page 21

## XML Support for ABM and LM in N3500

The following commands show XML Output for ABM and LM:

### show hardware profile buffer monitor sampling

#### CLI :

```
MTC-8(config)# show hardware profile buffer monitor sampling
Sampling CLI issued at: 05/25/2016 04:18:56
Sampling interval: 200
```

#### XML :

```
MTC-8(config)# show hardware profile buffer monitor sampling | xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://www.cisco.com/nxos:1.0:mtc_usd_cli">
 <nf:data>
 <show>
 <hardware>
 <profile>
 <buffer>
 <monitor>
 <__XML__BLK_Cmd_show_hardware_profile_buffer_monitor_summary>
 <__XML__OPT_Cmd_show_hardware_profile_buffer_monitor__readonly__>
 <__readonly__>
 <cmd_name>Sampling CLI</cmd_name>
```

```

<cmd_issue_time>05/25/2016 04:19:12</cmd_issue_time>
<TABLE_sampling>
 <ROW_sampling>
 <sampling_interval>200</sampling_interval>
 </ROW_sampling>
</TABLE_sampling>
</__readonly__>
</__XML__OPT_Cmd_show_hardware_profile_buffer_monitor__readonly__>
</__XML__BLK_Cmd_show_hardware_profile_buffer_monitor_summary>
</monitor>
</buffer>
</profile>
</hardware>
</show>
</nf:data>
</nf:rpc-reply>
]]>]]>

```

### show hardware profile buffer monitor detail | xml

**XML :**

```

<show>
 <hardware>
 <profile>
 <buffer>
 <monitor>
 <__XML__BLK_Cmd_show_hardware_profile_buffer_monitor_summary>
 <__XML__OPT_Cmd_show_hardware_profile_buffer_monitor__readonly__>
 <__readonly__>
 <cmd_name>Detail CLI</cmd_name>
 <cmd_issue_time>10/02/2001 10:58:58</cmd_issue_time>
 <TABLE_detail_entry>
 <ROW_detail_entry>
 <detail_util_name>Ethernet1/1</detail_util_name>
 <detail_util_state>Active</detail_util_state>
 </ROW_detail_entry>
 <ROW_detail_entry>
 <time_stamp>10/02/2001 10:58:58</time_stamp>
 <__XML__DIGIT384k_util>0</__XML__DIGIT384k_util>
 <__XML__DIGIT768k_util>0</__XML__DIGIT768k_util>
 <__XML__DIGIT1152k_util>0</__XML__DIGIT1152k_util>
 <__XML__DIGIT1536k_util>0</__XML__DIGIT1536k_util>
 <__XML__DIGIT1920k_util>0</__XML__DIGIT1920k_util>
 <__XML__DIGIT2304k_util>0</__XML__DIGIT2304k_util>
 <__XML__DIGIT2688k_util>0</__XML__DIGIT2688k_util>
 <__XML__DIGIT3072k_util>0</__XML__DIGIT3072k_util>
 <__XML__DIGIT3456k_util>0</__XML__DIGIT3456k_util>
 <__XML__DIGIT3840k_util>0</__XML__DIGIT3840k_util>
 <__XML__DIGIT4224k_util>0</__XML__DIGIT4224k_util>
 <__XML__DIGIT4608k_util>0</__XML__DIGIT4608k_util>
 <__XML__DIGIT4992k_util>0</__XML__DIGIT4992k_util>
 <__XML__DIGIT5376k_util>0</__XML__DIGIT5376k_util>
 <__XML__DIGIT5760k_util>0</__XML__DIGIT5760k_util>
 <__XML__DIGIT6144k_util>0</__XML__DIGIT6144k_util>
 </ROW_detail_entry>
 </__readonly__>
 </__XML__OPT_Cmd_show_hardware_profile_buffer_monitor__readonly__>
 </__XML__BLK_Cmd_show_hardware_profile_buffer_monitor_summary>
 </monitor>
 </buffer>
 </profile>
 </hardware>
</show>

```



```

<__XML__DIGIT5760k_util>0</__XML__DIGIT5760k_util>
<__XML__DIGIT6144k_util>0</__XML__DIGIT6144k_util>
</ROW_detail_entry>

```

### show hardware profile buffer monitor brief

**XML :**

```

show hardware profile buffer monitor brief | xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://www.cisco.com/nxos:1.0:mtc_usd_cli">
 <nf:data>
 <show>
 <hardware>
 <profile>
 <buffer>
 <monitor>
 <__XML__BLK_Cmd_show_hardware_profile_buffer_monitor_summary>
 <__XML__OPT_Cmd_show_hardware_profile_buffer_monitor__readonly__>
 <__readonly__>
 <cmd_name>Brief CLI</cmd_name>
 <cmd_issue_time>03/21/2016 09:06:38</cmd_issue_time>
 <TABLE_ucst_hdr>
 <ROW_ucst_hdr>
 <ucst_hdr_util_name>Buffer Block 1</ucst_hdr_util_name>
 <ucst_hdr_1sec_util>0KB</ucst_hdr_1sec_util>
 <ucst_hdr_5sec_util>0KB</ucst_hdr_5sec_util>
 <ucst_hdr_60sec_util>N/A</ucst_hdr_60sec_util>
 <ucst_hdr_5min_util>N/A</ucst_hdr_5min_util>
 <ucst_hdr_1hr_util>N/A</ucst_hdr_1hr_util>
 <ucst_hdr_total_buffer>Total Shared Buffer Available = 5397 Kbytes
 </ucst_hdr_total_buffer>
 <ucst_hdr_class_threshold>Class Threshold Limit = 5130 Kbytes
 </ucst_hdr_class_threshold>
 </ROW_ucst_hdr>
 </TABLE_ucst_hdr>
 <TABLE_brief_entry>
 <ROW_brief_entry>
 <brief_util_name>Ethernet1/45</brief_util_name>
 <brief_1sec_util>0KB</brief_1sec_util>
 <brief_5sec_util>0KB</brief_5sec_util>
 <brief_60sec_util>N/A</brief_60sec_util>
 <brief_5min_util>N/A</brief_5min_util>
 <brief_1hr_util>N/A</brief_1hr_util>
 </ROW_brief_entry>
 <brief_util_name>Ethernet1/46</brief_util_name>
 <brief_1sec_util>0KB</brief_1sec_util>
 <brief_5sec_util>0KB</brief_5sec_util>
 <brief_60sec_util>N/A</brief_60sec_util>
 <brief_5min_util>N/A</brief_5min_util>
 <brief_1hr_util>N/A</brief_1hr_util>
 </ROW_brief_entry>
 <brief_util_name>Ethernet1/47</brief_util_name>
 <brief_1sec_util>0KB</brief_1sec_util>
 <brief_5sec_util>0KB</brief_5sec_util>
 <brief_60sec_util>N/A</brief_60sec_util>
 <brief_5min_util>N/A</brief_5min_util>
 <brief_1hr_util>N/A</brief_1hr_util>
 </ROW_brief_entry>
 <brief_util_name>Ethernet1/48</brief_util_name>
 <brief_1sec_util>0KB</brief_1sec_util>
 <brief_5sec_util>0KB</brief_5sec_util>
 <brief_60sec_util>N/A</brief_60sec_util>
 <brief_5min_util>N/A</brief_5min_util>
 <brief_1hr_util>N/A</brief_1hr_util>
 </ROW_brief_entry>
 <brief_util_name>Ethernet1/21</brief_util_name>
 <brief_1sec_util>0KB</brief_1sec_util>
 <brief_5sec_util>0KB</brief_5sec_util>
 <brief_60sec_util>N/A</brief_60sec_util>
 <brief_5min_util>N/A</brief_5min_util>
 <brief_1hr_util>N/A</brief_1hr_util>
 </ROW_brief_entry>
 <brief_util_name>Ethernet1/22</brief_util_name>
 <brief_1sec_util>0KB</brief_1sec_util>
 <brief_5sec_util>0KB</brief_5sec_util>

```



```

<brief_60sec_util>N/A</brief_60sec_util>
<brief_5min_util>N/A</brief_5min_util>
<brief_1hr_util>N/A</brief_1hr_util>
<brief_util_name>Ethernet1/23</brief_util_name>
<brief_1sec_util>OKB</brief_1sec_util>
<brief_5sec_util>OKB</brief_5sec_util>

<brief_60sec_util>N/A</brief_60sec_util>
<brief_5min_util>N/A</brief_5min_util>
<brief_1hr_util>N/A</brief_1hr_util>
<brief_util_name>Ethernet1/24</brief_util_name>
<brief_1sec_util>OKB</brief_1sec_util>
<brief_5sec_util>OKB</brief_5sec_util>
<brief_60sec_util>N/A</brief_60sec_util>
<brief_5min_util>N/A</brief_5min_util>
<brief_1hr_util>N/A</brief_1hr_util>
<brief_util_name>Ethernet1/9</brief_util_name>
<brief_1sec_util>OKB</brief_1sec_util>
<brief_5sec_util>OKB</brief_5sec_util>
<brief_60sec_util>N/A</brief_60sec_util>
<brief_5min_util>N/A</brief_5min_util>
<brief_1hr_util>N/A</brief_1hr_util>
<brief_util_name>Ethernet1/10</brief_util_name>
<brief_1sec_util>OKB</brief_1sec_util>
<brief_5sec_util>OKB</brief_5sec_util>
<brief_60sec_util>N/A</brief_60sec_util>
<brief_5min_util>N/A</brief_5min_util>
<brief_1hr_util>N/A</brief_1hr_util>
<brief_util_name>Ethernet1/11</brief_util_name>
<brief_1sec_util>OKB</brief_1sec_util>
<brief_5sec_util>OKB</brief_5sec_util>
<brief_60sec_util>N/A</brief_60sec_util>
<brief_5min_util>N/A</brief_5min_util>
<brief_1hr_util>N/A</brief_1hr_util>
<brief_util_name>Ethernet1/12</brief_util_name>
<brief_1sec_util>OKB</brief_1sec_util>
<brief_5sec_util>OKB</brief_5sec_util>
<brief_60sec_util>N/A</brief_60sec_util>
<brief_5min_util>N/A</brief_5min_util>
<brief_1hr_util>N/A</brief_1hr_util>

```

### show hardware profile latency monitor sampling

#### CLI

```

MTC-8(config)# show hardware profile latency monitor sampling

Sampling CLI issued at: 05/25/2016 04:19:54

Sampling interval: 20

```

#### XML

```

MTC-8(config)# show hardware profile latency monitor sampling | xml

<?xml version="1.0" encoding="ISO-8859-1"?>

<nf:rpc-reply xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://www.cisco.com/nxos:1.0:mtc_usd_cli">

 <nf:data>

 <show>

 <hardware>

 <profile>

 <latency>

 <monitor>

 <__XML__BLK_Cmd_show_hardware_profile_latency_monitor_summary>

```

```

 <__XML__OPT_Cmd_show_hardware_profile_latency_monitor__readonly__>
 <__readonly__>
 <cmd_issue_time>05/25/2016 04:20:06</cmd_issue_time>
 <device_instance>0</device_instance>
 <TABLE_sampling>
 <ROW_sampling>
 <sampling_interval>20</sampling_interval>
 </ROW_sampling>
 </TABLE_sampling>
 </__readonly__>
 </__XML__OPT_Cmd_show_hardware_profile_latency_monitor__readonly__>
 </__XML__BLK_Cmd_show_hardware_profile_latency_monitor_summary>
</monitor>
</latency>
</profile>
</hardware>
</show>
</nf:data>
</nf:rpc-reply>
]]>]]>

```

### show hardware profile latency monitor threshold

#### CLI

```

MTC-8(config)# show hardware profile latency monitor threshold
Sampling CLI issued at: 05/25/2016 04:20:53
Threshold Avg: 3000
Threshold Max: 300000

```

#### XML

```

MTC-8(config)# show hardware profile latency monitor threshold | xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<nf:rpc-reply xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://ww.
cisco.com/nxos:1.0:mtc_usd_cli">
 <nf:data>
 <show>
 <hardware>
 <profile>
 <latency>
 <monitor>

```

```
<__XML__BLK_Cmd_show_hardware_profile_latency_monitor_summary>
 <__XML__OPT_Cmd_show_hardware_profile_latency_monitor__readonly__>
 <__readonly__>
 <cmd_issue_time>05/25/2016 04:21:04</cmd_issue_time>
 <device_instance>0</device_instance>
 <TABLE_threshold>
 <ROW_threshold>
 <threshold_avg>3000</threshold_avg>
 <threshold_max>300000</threshold_max>
 </ROW_threshold>
 </TABLE_threshold>
 </__readonly__>
 </__XML__OPT_Cmd_show_hardware_profile_latency_monitor__readonly__>
</__XML__BLK_Cmd_show_hardware_profile_latency_monitor_summary>
</monitor>
</latency>
</profile>
</hardware>
</show>
</nf:data>
</nf:rpc-reply>
]]>]]>
```





## INDEX

### B

- Bash [1, 3](#)
  - accessing [1](#)
  - examples [3](#)
  - feature bash-shell [1](#)
- Bourne-Again SHell, See [Bash](#)

### N

- NX-API [5, 6, 7, 9, 10, 11, 14, 19](#)
  - CLI [7](#)
  - cookie [6](#)
  - management commands [10](#)
  - message format [6](#)
  - request elements [11](#)
  - response codes [19](#)
  - response elements [14](#)
  - security [6](#)
  - transport [6](#)
  - user interface [9](#)

