



Converting CLI Commands to Network Configuration Format

This chapter explains how to install and use the XMLIN tool to convert CLI commands to the Network Configuration (NETCONF) protocol.

- [Finding Feature Information, on page 1](#)
- [Information About XMLIN, on page 1](#)
- [Installing and Using the XMLIN Tool, on page 2](#)
- [Converting Show Command Output to XML, on page 2](#)
- [Configuration Examples for XMLIN, on page 3](#)
- [Related Documents, on page 5](#)
- [Feature History for XMLIN, on page 5](#)

Finding Feature Information

Your software release might not support all the features documented in this module. For the latest caveats and feature information, see the Bug Search Tool at <https://tools.cisco.com/bugsearch> and the release notes for your software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the “New and Changed Information” chapter or the Feature History table.

Information About XMLIN

The XMLIN tool converts CLI commands to the Network Configuration (NETCONF) protocol format. NETCONF is a network management protocol that provides mechanisms to install, manipulate, and delete the configuration of network devices. It uses XML-based encoding for configuration data and protocol messages. The NX-OS implementation of the NETCONF protocol supports the following protocol operations: `<get>`, `<edit-config>`, `<close-session>`, `<kill-session>`, and `<exec-command>`.

The XMLIN tool converts show, EXEC, and configuration commands to corresponding NETCONF `<get>`, `<exec-command>`, and `<edit-config>` requests. You can enter multiple configuration commands into a single NETCONF `<edit-config>` instance.

The XMLIN tool also converts the output of show commands to XML format.

Installing and Using the XMLIN Tool

You can install the XMLIN tool and then use it to convert configuration commands to NETCONF format.

Before you begin

Although the XMLIN tool is usually capable of generating NETCONF instances of commands even if the corresponding feature sets or the required hardware capabilities are not available on the device, you might have to install some feature sets before entering the **xmlin** command.

Procedure

	Command or Action	Purpose
Step 1	switch# xmlin	
Step 2	switch(xmlin)# configure terminal	Enters global configuration mode.
Step 3	Configuration commands	Converts configuration commands to NETCONF format.
Step 4	(Optional) switch(config)(xmlin)# end	Generates the corresponding <edit-config> request. Note You must enter the end command to finish the current XML configuration before you generate an XML instance for a show command.
Step 5	(Optional) switch(config-if-verify)(xmlin)# show commands	Converts show commands to NETCONF format.
Step 6	(Optional) switch(config-if-verify)(xmlin)# exit	Returns to EXEC mode.

Converting Show Command Output to XML

You can convert the output of show commands to XML.

Before you begin

Make sure that all features for the commands you want to convert are installed and enabled on the device. Otherwise, the commands will fail.

You can use the **terminal verify-only** command to verify that a feature is enabled without entering it on the device.

Make sure that all required hardware for the commands you want to convert are present on the device. Otherwise, the commands will fail.

Make sure that the XMLIN tool is installed.

Procedure

	Command or Action	Purpose
Step 1	switch# <i>show-command</i> xmlin	Enters global configuration mode. Note You cannot use this command with configuration commands.

Configuration Examples for XMLIN

The following example shows how the XMLIN tool is installed on the device and used to convert a set of configuration commands to an <edit-config> instance.

```
switch# xmlin
*****
Loading the xmlin tool. Please be patient.
*****
Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Copyright © 2002-2013, Cisco Systems, Inc. All rights reserved.
The copyrights to certain works contained in this software are
owned by other third parties and used and distributed under
license. Certain components of this software are licensed under
the GNU General Public License (GPL) version 2.0 or the GNU
Lesser General Public License (LGPL) Version 2.1. A copy of each
such license is available at
http://www.opensource.org/licenses/gpl-2.0.php and
http://www.opensource.org/licenses/lgpl-2.1.php

switch(xmlin)# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)(xmlin)# interface ethernet 2/1
% Success
switch(config-if-verify)(xmlin)# cdp enable
% Success
switch(config-if-verify)(xmlin)# end
<?xml version="1.0"?>
<nf:rpc xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns="http://www.cisco.com/nxos:6.2.2.:configure_"
xmlns:m="http://www.cisco.com/nxos:6.2.2.:_exec"
xmlns:ml="http://www.cisco.com/nxos:6.2.2.:configure__if-eth-base" message-id="1">
  <nf:edit-config>
    <nf:target>
      <nf:running/>
    </nf:target>
    <nf:config>
      <m:configure>
        <m:terminal>
          <interface>
            <__XML_PARAM_interface>
              <__XML_value>Ethernet2/1</__XML_value>
              <ml:cdp>
                <ml:enable/>
              </ml:cdp>
            </__XML_PARAM_interface>
          </interface>
        </m:terminal>
      </m:configure>
    
```

```

        </nf:config>
    </nf:edit-config>
</nf:rpc>
]]>]]>

```

The following example shows how you must enter the **end** command to finish the current XML configuration before you generate an XML instance for a **show** command.

```

switch(xmlin)# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)(xmlin)# interface ethernet 2/1
switch(config-if-verify)(xmlin)# show interface ethernet 2/1
*****
Please type "end" to finish and output the current XML document before building a new one.
*****
% Command not successful

switch(config-if-verify)(xmlin)# end
<?xml version="1.0"?>
<nf:rpc xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns="http://www.cisco.com/nxos:6.2.2.:configure_"
xmlns:m="http://www.cisco.com/nxos:6.2.2.:_exec" message-id="1">
  <nf:edit-config>
    <nf:target>
      <nf:running/>
    </nf:target>
    <nf:config>
      <m:configure>
        <m:terminal>
          <interface>
            <__XML_PARAM__interface>
              <__XML_value>Ethernet2/1</__XML_value>
            </__XML_PARAM__interface>
          </interface>
        </m:terminal>
      </m:configure>
    </nf:config>
  </nf:edit-config>
</nf:rpc>
]]>]]>

switch(xmlin)# show interface ethernet 2/1
<?xml version="1.0"?>
<nf:rpc xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns="http://www.cisco.com/nxos:6.2.2.:if_manager" message-id="1">
  <nf:get>
    <nf:filter type="subtree">
      <show>
        <interface>
          <__XML_PARAM__ifeth>
            <__XML_value>Ethernet2/1</__XML_value>
          </__XML_PARAM__ifeth>
        </interface>
      </show>
    </nf:filter>
  </nf:get>
</nf:rpc>
]]>]]>
switch(xmlin)# exit
switch#

```

The following example shows how you can convert the output of the **show interface brief** command to XML.

```
switch# show interface brief | xmlin
<?xml version="1.0"?>
<nf:rpc xmlns:nf="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns="http://www.cisco.com/nxos:6.2.2.:if_manager"

message-id="1">
  <nf:get>
    <nf:filter type="subtree">
      <show>
        <interface>
          <brief/>
        </interface>
      </show>
    </nf:filter>
  </nf:get>
</nf:rpc>
]]>]]>
```

Related Documents

Related Topic	Document Title
XMLIN commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	<i>Cisco Nexus 7000 Series NX-OS System Management Command Reference</i>

Feature History for XMLIN

Your software release might not support all the features in this document. For the latest caveats and feature information, see the Bug Search Tool at <https://tools.cisco.com/bugsearch/> and the release notes for your software release.

Table 1: Feature History for XMLIN

Feature Name	Releases	Feature Information
XMLIN	6.2(2)	This feature was introduced.

