



CHAPTER 1

Overview

This chapter describes how the XML API is used to configure and monitor the Cisco Nexus 1000V, and includes the following sections:

- [Definition of Terms, page 1-1](#)
- [Introduction, page 1-1](#)
- [Schema, page 1-2](#)
- [Server, page 1-2](#)
- [Network Configuration Protocol, page 1-3](#)
- [XML Message and Reply Format, page 1-3](#)
- [Additional References, page 1-7](#)

Introduction

The eXtensible markup language (XML) application programming interface (API) lets you manage and monitor the Cisco Nexus 1000V using XML. From a client PC, you can encode CLI commands with XML API tags that are then sent to the device over a secure SSH connection.

Definition of Terms

[Table 1-1](#) defines the words, acronyms, and actions used throughout this guide.

Table 1-1 *Definition of Terms*

Term	Description
CLI	Command-line interface
NETCONF	Network Configuration Protocol
NETCONF operations	The actions carried out by NETCONF protocol.
RPC	Remote procedure call. A programming interface that allows your client PC to use the services of the remote XML server. Your client PC sends a request message to the remote XML server, which performs the operation requested, and replies with the results.
SSH	Secure Shell

Send document comments to nexus1k-docfeedback@cisco.com.

Table 1-1 Definition of Terms

Term	Description
XML	Extensible markup language, consisting mainly of text and tags.
XML agent	The XML server — An XML process on the Cisco Nexus 1000V responsible for carrying out client requests and then responding.
XML client	An external application that sends requests to, and receives responses from, the Cisco Nexus 1000V XML server.
XML operation	The portion of an XML client request that specifies what you want the XML agent to do.
XML operation provider	The Cisco Nexus 1000V code that does the following: <ul style="list-style-type: none"> • Parses XML requests. • Carries out the requested operations. • Assembles the XML reply.
XML request	An XML document sent to the Cisco Nexus 1000V containing a number of requested operations to be carried out.
XML reply	The reply to an XML request.
XML schema	An XML document specifying the structure and possible contents of XML elements that can be contained in an XML document.

Schema

The XML interface is implemented with an XML schema definition (XSD) of the supported CLI commands in XML. You can download the feature-based XSD files from Cisco.com.

To obtain a copy of the XSD, from your browser, navigate to the Cisco software download site at the following URL:

<http://www.cisco.com/go/1000v/>

Server

A Cisco Nexus 1000V XML agent, also called the XML server, enables configuration and monitoring using an exchange of XML formatted request and reply streams over a secure connection.

When you start an SSH session with the XML server, it sends an immediate hello message including its capabilities, as shown in [Example 1-1](#). Before the server processes further requests, you must advertise your capabilities in a hello message to the server, as shown in [Example 1-2](#).



Note

You must end all XML documents with `]]>]]>` to support synchronization in NETCONF over SSH.

Example 1-1 Hello Message from the Server

```
<?xml version="1.0"?>
<hello xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <capabilities>
    <capability>urn:ietf:params:xml:ns:netconf:base:1.0</capability>
```

Send document comments to nexus1k-docfeedback@cisco.com.

```
</capabilities>  
</hello>]]>]]>
```

Example 1-2 Hello Message from the Client

```
<?xml version="1.0"?>  
<nc:hello xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">  
  <nc:capabilities>  
    <nc:capability>urn:ietf:params:xml:ns:netconf:base:1.0</nc:capability>  
  </nc:capabilities>  
</nc:hello>]]>]]>
```

Network Configuration Protocol

Communication with the XML API is accomplished in XML over the Network Configuration Protocol (NETCONF). NETCONF uses a simple remote procedure call (RPC) and SSH transport protocol for a secure connection.

To run NETCONF over SSHv2, the client establishes an SSH transport connection with the XML server. The client and server exchange keys for security and password encryption. The NETCONF SSHv2 session user ID and password are used for authorization and authentication. The user privilege level is enforced and the client session may not have full access to the NETCONF operations if the privilege level is not high enough.

If AAA is configured, the AAA service is used as if you had established an SSH session directly to the Cisco Nexus 1000V. Once the client is successfully authenticated, it starts the SSH connection protocol and the SSH session is established. After the SSH session is established, the user or application starts NETCONF as an SSH subsystem.

For detailed information about NETCONF, see [RFC 4741](#).

For detailed information about using the NETCONF protocol over the Secure Shell (SSH), see [RFC 4742](#).

XML Message and Reply Format

The communication between client and server consists of a series of alternating request and reply messages.

A valid client ID is an unsigned 32-bit integer. If the request message contains a valid client ID, then the server also includes the same client ID in its reply. The client ID is treated as opaque data and ignored in every other way by the XML API.

Every request sent by a client application to the server begins with an XML declaration tag followed by a request tag and one or more operation tags. Each request can contain one or more operations for each supported operation type, and the operations can be repeated.

Similarly, every [reply](#) from the server begins with an XML declaration tag followed by a reply tag and one or more operation tags corresponding to those in the client request.

[Figure 1-1](#) shows the tags used to identify each line of content in NETCONF XML messages:

- [XML Declaration, page 1-4](#)
- [Request, page 1-4](#)
- [Operations, page 1-5](#)

Send document comments to nexus1k-docfeedback@cisco.com.

Figure 1-1 NETCONF XML Message Format

XML Declaration	→	<?xml version="1.0"?>
*Request Start	→	<nc:rpc message-id="1" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns="http://www.cisco.com/nxos:1.0:xml">
*NETCONF Operation Start	→	<nc:get>
*Device Operation Start	→	<nc:filter type="subtree">
		<show>
		<xml>
		<server>
Device Operation End	→	<status/>
		</server>
		</xml>
		</show>
NETCONF Operation End	→	</nc:filter>
		</nc:get>
Request End	→	</nc:rpc>]]>]]>

***For every start tag in an XML message, there must be a corresponding end tag.**

XML Declaration

Each request and reply begins with an XML declaration indicating which version of XML and (optionally) which character set are being used.

The following are the attributes of the XML declaration:

- Version—Specifies the version of XML to be used.
- Encoding—(optional) Specifies the standardized character set to be used.

Example 1-3 Declaration Statement

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

Request

In the XML message format, the client request follows the XML declaration tag. Each request must be enclosed within a set of RPC tags. Requests include NETCONF and device (CLI-based) operations.

As shown in [Example 1-4](#), you first specify the NETCONF operation (for example, **get**) followed by the CLI commands (for example, **show xml server status**).

Example 1-4 Request

```
<nc:rpc message-id="1" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
  xmlns="http://www.cisco.com/nxos:1.0:xml">
  <nc:get>
    <nc:filter type="subtree">
      <show>
        <xml>
          <server>
```

Send document comments to nexus1k-docfeedback@cisco.com.

```
<status/>
</server>
</xml>
</show>
</nc:filter>
</nc:get>
</nc:rpc>]]>]]>
```

Operations

In your client request, you specify both NETCONF operations, and CLI command-based operations. [Example 1-5](#) shows how the following operations appear in an XML API message.

NETCONF Operation	CLI Command Operation
get	
filter type = subtree	show xml server status



Note

To identify which of the NETCONF operations are supported, see [Table 1-2](#).

Example 1-5 Operations

```
<nc:get>
  <nc:filter type="subtree">
    <show>
      <xml>
        <server>
          <status/>
        </server>
      </xml>
    </show>
  </nc:filter>
</nc:get>
```

When your client request is received by an XML agent, the request is routed to the XML API library for processing. After all requested operations are processed, the XML agent sends an XML encoded reply stream back to your client. [Table 1-2](#) shows which of the NETCONF operations are supported.

Table 1-2 NETCONF Operations

NETCONF Operation	Supported?	Description	Example
close-session	Yes	Terminates your server session.	Example: Closing Your Session, page 3-10
commit	No	—	—
copy-config	No	—	—

Send document comments to nexus1k-docfeedback@cisco.com.

Table 1-2 **NETCONF Operations (continued)**

NETCONF Operation	Supported?	Description	Example
delete-config	Yes	Performs the equivalent of the write erase command on the startup configuration.	Example: Deleting a Configuration, page 3-9
edit-config	Yes	Configures features in the running configuration of the device. You use this operation for configuration commands.	Example: Editing a Configuration, page 3-11
get	Yes	Receives configuration information from the device. You use this operation for show commands. The source of the data is the running configuration.	Example: Creating a Message with Multiple Operations, page 3-5
get-config	No	—	—
kill-session	Yes	Terminates another server session.	Example: Closing a Session Other Than Your Own, page 3-10
lock	No	—	—
unlock	No	—	—
validate	No	—	—

Reply

For every XML request you send from the client, the XML server sends an XML reply, as shown in [Example 1-6](#). Just as in your request message, every reply from the server begins with an [XML declaration](#) and includes one or more [operations](#). The possible replies are described in [Table 1-3](#).



Note

Replies may be received in a different sequence than the requests were sent.

Example 1-6 Reply

```
<nc:rpc-reply message-id="315" xmlns xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</nc:rpc-reply>]]]]>
```

Table 1-3 **Replies**

Element	Description
<ok>	The request completed successfully and no additional information is included in the reply.
<data>	The request completed successfully and additional information is included in the reply.
<rpc-error>	The request failed, and error information is included with the reply.

Send document comments to nexus1k-docfeedback@cisco.com.

Additional References

For additional information related to implementing the XML management interface, see the following sections:

- [Standards, page 1-7](#)
- [RFCs, page 1-7](#)

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

RFCs

RFCs	Title
RFC 4741	<i>NETCONF Configuration Protocol</i>
RFC 4742	<i>Using the NETCONF Configuration Protocol over Secure SHell (SSH)</i>

Send document comments to nexus1k-docfeedback@cisco.com.