Cisco Nexus 7000 Series NX-OS Fundamentals Configuration Guide, Release 5.x

First Published: July 31, 2009

Last Modified: June 20, 2011

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

Text Part Number: OL-19602-02

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: http://www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2013 Cisco Systems, Inc. All rights reserved.

CONTENTS

Preface

This preface describes the audience, organization and conventions of the *Cisco Nexus 5000 Series NX-OS Fundamentals Configuration Guide*. It also provides information on how to obtain related documentation.

Audience

This publication is for experienced users who configure and maintain Cisco NX-OS devices.

Document Organization

This document is organized into the following chapters:

Chapter	Description
New and Changed Information, on page 1	Describes the new and changed information for the new Cisco NX-OS software release.
Overview, on page 5	Provides an overview of the features included in the Cisco NX-OS software.
Using the Cisco NX-OS Setup Utility, on page 19	Provides a flowchart for setting up the Cisco NX-OS software.
Understanding the Command-Line Interface, on page 27	Describes the command line interface including command modes, special characters, and keystrokes.

Chapter	Description
Configuring Terminal Settings and Sessions, on page 61	Describes how to manage the terminal settings and sessions on a Cisco NX-OS device.
Basic Device Management, on page 77	Describes basic system management, including setting the clock and configuring a message of the day.
Using the Device File Systems, Directories, and Files, on page 87	Describes how to configure and manage file systems, directories, and files on an Cisco NX-OS device.
Working with Configuration Files, on page 105	Describes the configuration files and how to manage them.
Scripting with Tcl, on page 119	Describes how to run tcl interactively and in scripts.

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element(keyword or argument).
[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.
`variable`	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
`screen font`	Terminal sessions and information the switch displays are in screen font.
`boldface screen font`	Information you must enter is in boldface screen font.
italic screen font	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

Cisco NX-OS documentation is available at the following URL:

http://www.cisco.com/en/US/products/ps9372/tsd_products_support_series_home.html

The documentation set for the Cisco NX-OS software includes the following documents:

Release Notes

Cisco Nexus 7000 Series NX-OS Release Notes, Release 5.x

NX-OS Configuration Guides

Cisco Nexus 7000 Series NX-OS Configuration Examples, Release 5.x

Cisco Nexus 7000 Series NX-OS FabricPath Configuration Guide, Release 5.x

Cisco Nexus 7000 Series NX-OS Fundamentals Configuration Guide, Release 5.x

Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide, Release 5.x

Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide

Cisco Nexus 7000 Series NX-OS Multicast Routing Configuration Guide

Cisco Nexus 7000 Series NX-OS OTV Configuration Guide

Cisco Nexus 7000 Series NX-OS Quality of Service Configuration Guide, Release 5.x

Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 5.x

Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 5.x

Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 5.x

Cisco Nexus 7000 Series NX-OS Verified Scalability Guide

Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide

Cisco Nexus 7000 Series NX-OS Virtual Device Context Quick Start

Cisco Nexus 7000 Series OTV Quick Start Guide

Configuring Feature Set for FabricPath

Configuring the Cisco Nexus 2000 Series Fabric Extender

NX-OS Command References

Cisco Nexus 7000 Series NX-OS Command Reference Master Index

Cisco Nexus 7000 Series NX-OS FabricPath Command Reference

Cisco Nexus 7000 Series NX-OS Fundamentals Command Reference

Cisco Nexus 7000 Series NX-OS High Availability and Redundancy Command Reference

Cisco Nexus 7000 Series NX-OS Interfaces Command Reference

Cisco Nexus 7000 Series NX-OS Layer 2 Switching Command Reference

Cisco Nexus 7000 Series NX-OS Multicast Routing Command Reference

Cisco Nexus 7000 Series NX-OS OTV Command Reference

Cisco Nexus 7000 Series NX-OS Quality of Service Command Reference

Cisco Nexus 7000 Series NX-OS Security Command Reference

Cisco Nexus 7000 Series NX-OS System Management Command Reference

Cisco Nexus 7000 Series NX-OS Unicast Routing Command Reference

Cisco Nexus 7000 Series NX-OS Virtual Device Context Command Reference

Other Software Document

Cisco Nexus 7000 Series NX-OS High Availability and Redundancy Guide

Cisco Nexus 7000 Series NX-OS MIB Quick Reference

Cisco Nexus 7000 Series NX-OS Software Upgrade and Downgrade Guide, Release 5.x

Cisco Nexus 7000 Series NX-OS Troubleshooting Guide

Cisco NX-OS Licensing Guide

Cisco NX-OS System Messages Reference

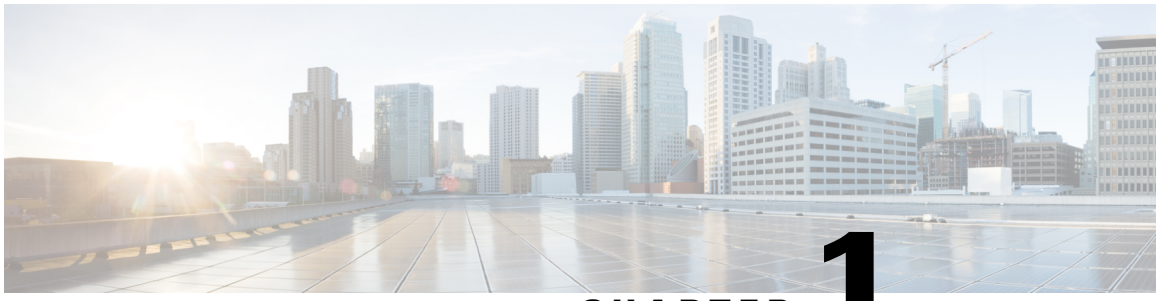
Cisco NX-OS XML Interface User Guide

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS version 2.0.



CHAPTER 1

New and Changed Information

This chapter provides release-specific information for each new and changed feature in the *Cisco Nexus 7000 Series NX-OS Fundamentals Configuration Guide, Release 5.x*. The latest version of this document is available at the following Cisco website:

http://www.cisco.com/en/US/products/ps9402/products_installation_and_configuration_guides_list.html

New and Changed Information

To check for additional information about Cisco NX-OS Release 4.2, see the *Cisco Nexus 7000 Series NX-OS Release Notes, Release 5.x* available at the following Cisco website:

http://www.cisco.com/en/US/products/ps9402/prod_release_notes_list.html

This table summarizes the new and changed features for the *Cisco Nexus 7000 Series NX-OS Fundamentals Configuration Guide, Release 5.x*, and tells you where they are documented.

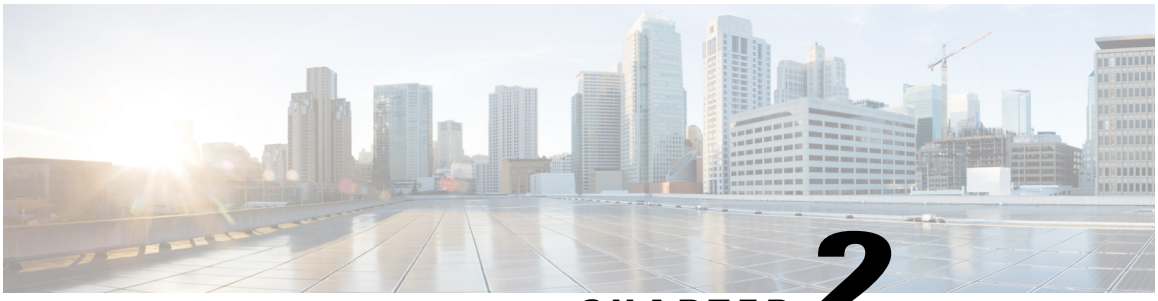
Table 1: New and Changed Features for Release 4.2

Feature	Description	Changed in Release	Where Documented
Filtering and searching utilities	Changes keywords from Linux style to Cisco NX-OS style.	4.0(3)	Understanding the Command-Line Interface, on page 27
grep and egrep utilities	Changes keywords from Linux style to Cisco NX-OS style.	4.0(3)	Understanding the Command-Line Interface, on page 27
Text echoing	Changed echo command syntax from Linux style keyword to Cisco NX-OS style keyword.	4.0(3)	Understanding the Command-Line Interface, on page 27

Feature	Description	Changed in Release	Where Documented
Persistent CLI variables	Added support for CLI variables that persist Persistent CLI variables across device reloads.	4.1(2)	Understanding the Command-Line Interface, on page 27
showcommand output filtering	Added the end keyword to filter show command output.	4.1(2)	Understanding the Command-Line Interface, on page 27
diff utility	Allows comparison of command outputs.	4.2(1)	Understanding the Command-Line Interface, on page 27
Command aliases	Can be used in **show** command searching and filtering. Allows command aliases for users sessions.	4.2(1)	Understanding the Command-Line Interface, on page 27
I/O module commands	Allows sending commands directly to a module from the supervisor module session.	4.2(1)	Understanding the Command-Line Interface, on page 27
Command history	Provides changes to the **show cli history** command.	4.2(1)	Understanding the Command-Line Interface, on page 27
Command modes	Allows saving and restoring of command modes.	4.2(1)	Understanding the Command-Line Interface, on page 27
Confirmation prompts	Allows enabling and disabling for command confirmation prompts.	4.2(1)	Understanding the Command-Line Interface, on page 27
Terminal colors	Allowed changes to the colors used for CLI elements in the terminal display.	4.2(1)	Understanding the Command-Line Interface, on page 27
show command output redirection	You can change the format of the **show** command output when you redirect it to a file. The format can be ASCII or zipped.	4.2(1)	Using the Device File Systems, Directories, and Files, on page 87

New and Changed Information

New and Changed Information



CHAPTER 2

Overview

This chapter provides an overview of the Cisco NX-OS software.

Software Compatibility

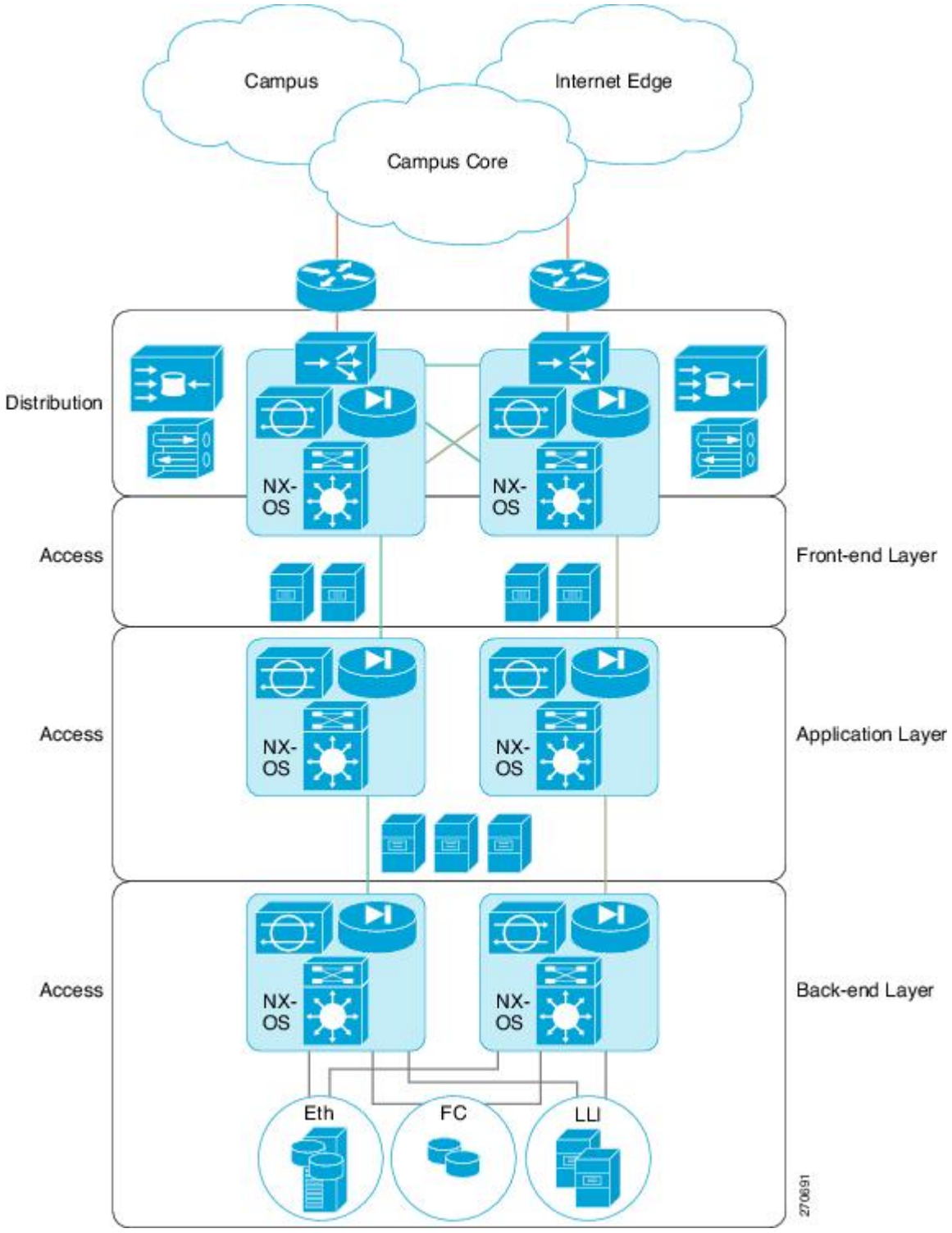
The Cisco NX-OS software interoperates with Cisco products that run any variant of the Cisco IOS software. The Cisco NX-OS software also interoperates with any networking operating system that conforms to the IEEE and RFC compliance standards.

Common Software Throughout the Data Center

The Cisco NX-OS software provides a unified operating system that is designed to run all areas of the data center network including the LAN and Layer 4 through Layer 7 network services.

This figure shows an overview of the Cisco NX-OS software in the data center.

Figure 1: Cisco NX-OS in a Data Center



Modular Software Design

The Cisco NX-OS software supports distributed multithreaded processing on symmetric multiprocessors (SMPs), multi-core CPUs, and distributed data module processors. The Cisco NX-OS software offloads computationally intensive tasks, such as hardware table programming, to dedicated processors distributed across the data modules. The modular processes are created on demand, each in a separate protected memory space. Processes are started and system resources are allocated only when you enable a feature. A real-time preemptive scheduler helps to ensure the timely processing of critical functions.

Virtual Device Contexts

The Cisco NX-OS software can segment system and hardware resources into virtual contexts that emulate virtual devices. Each virtual device context (VDC) has its own software processes, dedicated hardware resources (interfaces), and an independent management environment. With VDCs, you can consolidate separate networks onto a common infrastructure, which maintain the administrative boundary separation and fault isolation characteristics of physically separate networks, and provide many of the operational cost benefits of a single infrastructure. For more information, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide*.

Serviceability

The Cisco NX-OS software has serviceability functions that allow the device to respond to network trends and events. These features help you with network planning and improving response times.

Switched Port Analyzer

The Switched Port Analyzer (SPAN) feature allows you to analyze all traffic between ports (called the SPAN source ports) by nonintrusively directing the SPAN session traffic to a SPAN destination port that has an external analyzer attached to it. For more information about SPAN, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 5.x*.

Ethanalyzer

Ethanalyzer is a Cisco NX-OS protocol analyzer tool based on the Wireshark (formerly Ethereal) open source code. Ethanalyzer is a command-line version of Wireshark for capturing and decoding packets. You can use Ethanalyzer to troubleshoot your network and analyze the control-plane traffic. For more information about Ethanalyzer, see the *Cisco Nexus 7000 Series NX-OS Troubleshooting Guide -- Troubleshooting Tools and Methodology*.

Call Home

The Call Home feature continuously monitors hardware and software components to provide e-mail-based notification of critical system events. A versatile range of message formats is available for optimal compatibility with pager services, standard e-mail, and XML-based automated parsing applications. It offers alert grouping

capabilities and customizable destination profiles.You can use this feature, for example, to directly page a network support engineer, send an e-mail message to a network operations center (NOC), and employ Cisco AutoNotify services to directly generate a case with the Cisco Technical Assistance Center (TAC). For more information about Call Home, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 5.x*.

Online Diagnostics

Cisco generic online diagnostics (GOLD) verify that hardware and internal data paths are operating as designed. Boot-time diagnostics, continuous monitoring, and on-demand and scheduled tests are part of the Cisco GOLD feature set. GOLD allows rapid fault isolation and continuous system monitoring. For information about configuring GOLD, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 5.x*.

Embedded Event Manager

Cisco Embedded Event Manager (EEM) is a device and system management feature that helps you to customize behavior based on network events as they happen. For information about configuring EEM, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 5.x*.

NetFlow

The Cisco NX-OS NetFlow implementation supports version 5 and version 9 exports. It also supports the Flexible NetFlow configuration model and hardware-based Sampled NetFlow for enhanced scalability. For more information about NetFlow, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 5.x*.

Manageability

This section describes the manageability features in the Cisco NX-OS software.

Simple Network Management Protocol

The Cisco NX-OS software is compliant with Simple Network Management Protocol (SNMP) version 1, version 2, and version 3. A large number of MIBs is supported. For more information about SNMP, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 5.x*.

Configuration Verification and Rollback

The Cisco NX-OS software allows you to verify the consistency of a configuration and the availability of necessary hardware resources prior to committing the configuration. You can preconfigure a device and apply the verified configuration at a later time. Configurations also include checkpoints that allow you to roll back to a known good configuration as needed. For more information about rollback, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 5.x*.

Role-Based Access Control

With role-based access control (RBAC), you can limit access to device operations by assigning roles to users. You can customize access and restrict it to the users who require it. For more information about RBAC, see the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 5.x*.

Connectivity Management Processor

The Cisco NX-OS software supports the use of a Connectivity Management Processor (CMP) for remote platform management. The CMP provides an out-of-band access channel to the Cisco NX-OS console. For more information about CMP, see the *Cisco Nexus 7000 Series Connectivity Management Processor Configuration Guide*.

Cisco NX-OS Device Configuration Methods

You can configure devices using the CLI from a Secure Shell (SSH) session or a Telnet session. SSH provides a secure connection to the device. The CLI configuration guides and command references are organized by feature. For more information, see the Cisco NX-OS configuration guides and the Cisco NX-OS command references. For more information on SSH and Telnet, see the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 5.x*.

You can also configure devices using the XML management interface, which is a programmatic method based on the NETCONF protocol that complements the CLI. For more information, see the *Cisco NX-OS XML Interface User Guide* .

Traffic Routing, Forwarding, and Management

This section describes the traffic routing, forwarding, and management features supported by the Cisco NX-OS software.

Ethernet Switching

The Cisco NX-OS software supports high-density, high-performance Ethernet systems and provides the following Ethernet switching features:

- IEEE 802.1D-2004 Rapid and Multiple Spanning Tree Protocols (802.1w and 802.1s)
- IEEE 802.1Q VLANs and trunks
- 16,000-subscriber VLANs
- IEEE 802.3ad link aggregation
- Private VLANs
- Cross-chassis private VLANs
- Unidirectional Link Detection (UDLD) in aggressive and standard modes

For more information, see the *Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide, Release 5.x* and the *Cisco Nexus 7000 Series NX-OS Layer 2 Switching Configuration Guide*.

IP Routing

The Cisco NX-OS software supports IP version 4 (IPv4) and IP version 6 (IPv6) and the following routing protocols:

- Open Shortest Path First (OSPF) Protocol Versions 2 (IPv4) and 3 (IPv6)
- Intermediate System-to-Intermediate System (IS-IS) Protocol
- Border Gateway Protocol (BGP)
- Enhanced Interior Gateway Routing Protocol (EIGRP)
- Routing Information Protocol Version 2 (RIPv2)

The Cisco NX-OS software implementations of these protocols are fully compliant with the latest standards and include 4-byte autonomous system numbers (ASNs) and incremental shortest path first (SPF). All unicast protocols support Non-Stop Forwarding Graceful Restart (NSF-GR). All protocols support all interface types, including Ethernet interfaces, VLAN interfaces, subinterfaces, port channels, tunnel interfaces, and loopback interfaces.

For more information, see the *Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 5.x*.

IP Services

The following IP services are available in the Cisco NX-OS software:

- Virtual Routing and Forwarding (VRF)
- Dynamic Host Configuration Protocol (DHCP) Helper
- Hot-Standby Routing Protocol (HSRP)
- Gateway Load Balancing Protocol (GLBP)
- Enhanced Object Tracking
- Policy-Based Routing (PBR)
- Unicast Graceful Restart for all protocols in IPv4 Unicast Graceful Restart for OPSFv3 in IPv6

For more information, see the *Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 5.x*.

IP Multicast

The Cisco NX-OS software includes the following multicast protocols and functions:

- Protocol Independent Multicast (PIM) Version 2 (PIMv2)
- Source Specific Multicast (SSM)

- PIM sparse mode (Any-Source Multicast [ASM] for IPv4 and IPv6)

Note The Cisco NX-OS software does not support PIM dense mode.

- Bidirectional Protocol Independent Multicast (Bidir PIM)
- Anycast rendezvous point (Anycast-RP)
- Multicast NSF for IPv4 and IPv6
- RP-Discovery using bootstrap router (BSR) (Auto-RP and static)
- Internet Group Management Protocol (IGMP) Versions 1, 2, and 3 router role
- IGMPv2 host mode
- IGMP snooping
- Multicast Listener Discovery (MLD) Protocol Version 2 (for IPv6)
- Multicast Source Discovery Protocol (MSDP) (for IPv4 only)

For more information, see the *Cisco Nexus 7000 Series NX-OS Multicast Routing Command Reference*.

Quality of Service

The Cisco NX-OS software supports quality of service (QoS) functions for classification, marking, queuing, policing, and scheduling. Modular QoS CLI (MQC) supports all QoS features. You can use MQC to provide uniform configurations across various Cisco platforms. For more information, see the *Cisco Nexus 7000 Series NX-OS Quality of Service Configuration Guide, Release 5.x*.

Network Security

This section describes the network security features support by the Cisco NX-OS software.

Cisco TrustSec

Cisco TrustSec security provides data confidentiality and integrity and supports standard IEEE 802.1AE link-layer cryptography with 128-bit Advanced Encryption Standard (AES) cryptography. Link-layer cryptography guarantees end-to-end data privacy while allowing the insertion of security service devices along the encrypted path. Cisco TrustSec uses security group access control lists (SGACLs), which are based on security group tags instead of IP addresses. SGACLs enable policies that are more concise and easier to manage due to their topology independence. For more information, see the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 5.x*.

Additional Network Security Features

In addition to Cisco TrustSec, the Cisco NX-OS software includes the following security features:

- Data path intrusion detection system (IDS) for protocol conformance checks
- Control Plane Policing (CoPP)
- Message-digest algorithm 5 (MD5) routing protocol authentication
- Cisco-integrated security features, including Dynamic Address Resolution Protocol (ARP) inspection (DAI), DHCP snooping, and IP Source Guard
- Authentication, authorization, and accounting (AAA)
- RADIUS and TACACS+
- SSH Protocol Version 2
- SNMPv3
- Port security
- IEEE 802.1X authentication
- Layer 2 Cisco Network Admission Control (NAC) LAN port IP
- Policies based on MAC and IPv4 addresses supported by named ACLs (port-based ACLs [PACLs], VLAN-based ACLs [VACLs], and router-based ACLs [RACLs])
- Traffic storm control (unicast, multicast, and broadcast)
- Unicast Reverse Path Forwarding (Unicast RPF)

For more information, see the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 5.x*.

Licensing

The Cisco NX-OS software licensing feature allows you to access premium features on the device after you install the appropriate license for that feature. Any feature not included in a license package is bundled with the Cisco NX-OS software and is provided to you at no extra charge.

You must purchase and install a license for each device.

Note With the exception of the Cisco TrustSec feature, you can enable a feature without installing its license. The Cisco NX-OS software gives you a grace period that allows you to try a feature before purchasing its license. You must install the Advanced Services license package to enable the Cisco TrustSec feature.

For detailed information about Cisco NX-OS software licensing, see the *Cisco NX-OS Licensing Guide*.

For information about troubleshooting licensing issues, see the Cisco Nexus 7000 Series NX-OS Troubleshooting Guide.

Supported Standards

This table lists the IEEE compliance standards.

Table 2: IEEE Compliance Standards

Standard	Description
802.1D	MAC Bridges
802.1s	Multiple Spanning Tree Protocol
802.1w	Rapid Spanning Tree Protocol
802.1AE	MAC Security (link layer cryptography)
802.3ad	Link aggregation with LACP
802.3ab	1000BaseT (10/100/1000 Ethernet over copper)
802.3ae	10-Gigabit Ethernet
802.1Q	VLAN Tagging
802.1p	Class of Service Tagging for Ethernet frames
802.1X	Port-based network access control

This table lists the RFC compliance standards.

Table 3: RFC Compliance Standards

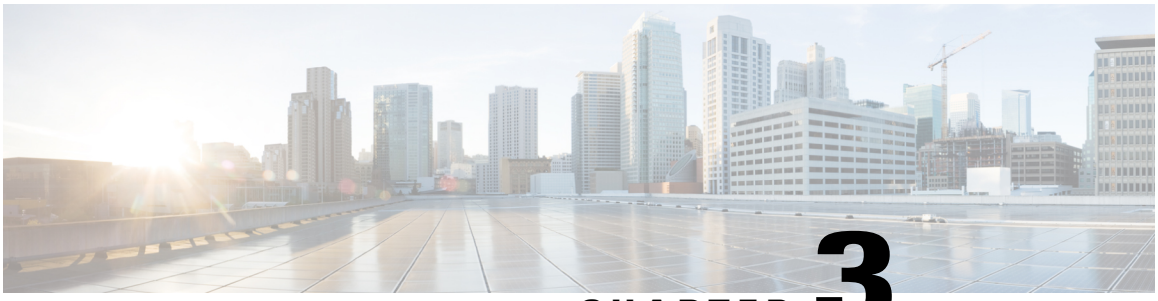
Standard	Description
BGP	
RFC 1997	BGP Communities Attribute
RFC 2385	Protection of BGP Sessions via the TCP MD5 Signature Option
RFC 2439	BGP Route flap damping
RFC 2519	A Framework for Inter-Domain Route Aggregation
RFC 2858	Multiprotocol Extensions for BGP-4
RFC 3065	Autonomous System Confederations for BGP
RFC 3392	Capabilities Advertisement with BGP-4

Standard	Description
RFC 4271	BGP version 4
RFC 4273	BGP4 MIB - Definitions of Managed Objects for BGP-4
RFC 4456	BGP Route reflection
RFC 4486	Subcodes for BGP cease notification message
RFC 4724	Graceful Restart Mechanism for BGP
RFC 4893	BGP Support for Four-octet AS Number Space
ietf-draft	Bestpath transition avoidance (draft-ietf-idr-avoid-transition-05.txt)
ietf-draft	Peer table objects (draft-ietf-idr-bgp4-mib-15.txt)
ietf-draft	Dynamic Capability (draft-ietf-idr-dynamic-cap-03.txt)
OSPF	
RFC 2370	OSPF Opaque LSA Option
RFC 2328	OSPF Version 2
RFC 2740	OSPF for IPv6 (OSPF version 3)
RFC 3101	OSPF Not-So-Stubby-Area (NSSA) Option
RFC 3137	OSPF Stub Router Advertisement
RFC 3509	Alternative Implementations of OSPF Area Border Routers
RFC 3623	Graceful OSPF Restart
RFC 4750	OSPF Version 2 MIB
RIP	
RFC 1724	RIPv2 MIB extension

Standard	Description
RFC 2082	RIPv2 MD5 Authentication
RFC 2453	RIP Version 2
IS-IS	
RFC 1142 (OSI 10589)	OSI 10589 Intermediate system to intermediate system intra-domain routing exchange protocol
RFC 1195	Use of OSI IS-IS for routing in TCP/IP and dual environment.
RFC 2763	Dynamic Hostname Exchange Mechanism for IS-IS
RFC 2966	Domain-wide Prefix Distribution with Two-Level IS-IS
RFC 2973	IS-IS Mesh Groups
RFC 3277	IS-IS Transient Blackhole Avoidance
RFC 3373	Three-Way Handshake for IS-IS Point-to-Point Adjacencies
RFC 3567	IS-IS Cryptographic Authentication
RFC 3847	Restart Signaling for IS-IS
ietf-draft	Internet Draft Point-to-point operation over LAN in link-state routing protocols (draft-ietf-isis-igp-p2p-over-lan-06.txt)
IP Services	
RFC 768	UDP
RFC 783	TFTP
RFC 791	IP
RFC 792	ICMP
RFC 793	TCP
RFC 826	ARP

Standard	Description
RFC 854	Telnet
RFC 959	FTP
RFC 1027	Proxy ARP
RFC 1305	NTP v3
RFC 1519	CIDR
RFC 1542	BootP relay
RFC 1591	DNS client
RFC 1812	IPv4 routers
RFC 2131	DHCP Helper
RFC 2338	VRRP
RFC 2784	Generic Routing Encapsulation (GRE)
IP-Multicast	
RFC 2236	Internet Group Management Protocol, Version 2
RFC 2710	Multicast Listener Discovery (MLD) for IPv6
RFC 3376	Internet Group Management Protocol, Version 3
RFC 3446	Anycast Rendezvous Point (RP) mechanism using Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP)
RFC 3569	An Overview of Source-Specific Multicast (SSM)
RFC 3618	Multicast Source Discovery Protocol (MSDP)
RFC 3810	Multicast Listener Discovery Version 2 (MLDv2) for IPv6

Standard	Description
RFC 4601	ASM - Sparse Mode (PIM-SM): Protocol Specification (Revised)
RFC 4607	Source-Specific Multicast for IP
RFC 4610	Anycast-RP Using Protocol Independent Multicast (PIM)
ietf-draft	Mtrace server functionality, to process mtrace-requests, draft-ietf-idmr-traceroute-ipm-07.txt
ietf-draft	Bi-directional Protocol Independent Multicast (BIDIR-PIM), draft-ietf-pim-bidir-09.txt



CHAPTER 3

Using the Cisco NX-OS Setup Utility

This chapter contains the following sections:

Information About the Cisco NX-OS Setup Utility

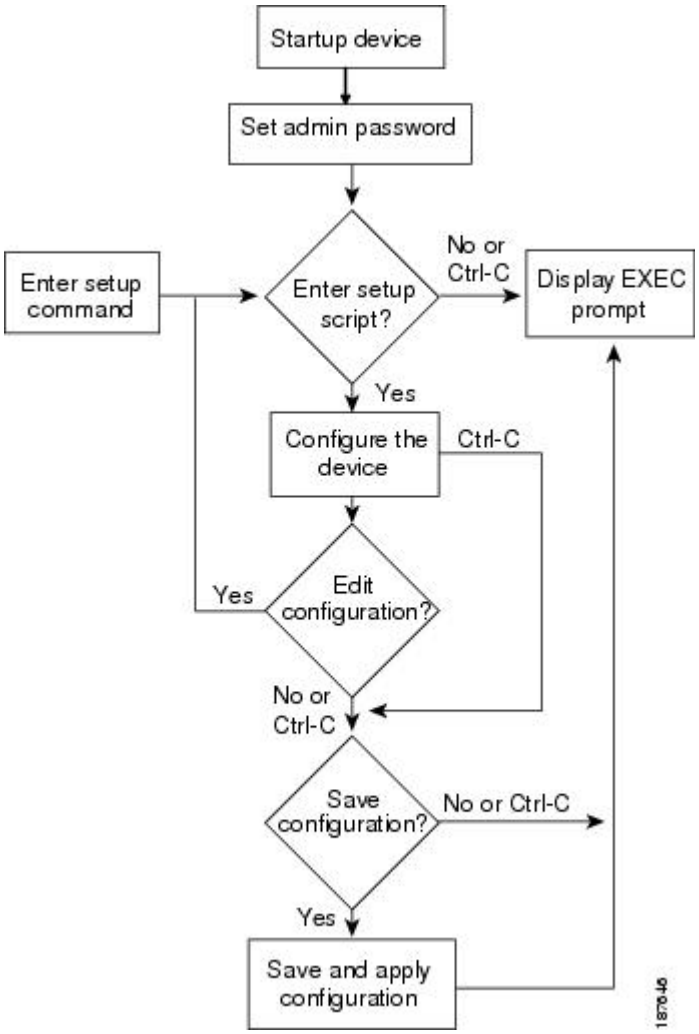
The Cisco NX-OS setup utility is an interactive command-line interface (CLI) mode that guides you through a basic (also called a startup) configuration of the system. The setup utility allows you to configure only enough connectivity for system management.

The setup utility allows you to build an initial configuration file using the System Configuration Dialog. The setup starts automatically when a device has no configuration file in NVRAM. The dialog guides you through initial configuration. After the file is created, you can use the CLI to perform additional configuration.

You can press **Ctrl-C** at any prompt to skip the remaining configuration options and proceed with what you have configured up to that point, except for the administrator password. If you want to skip answers to any questions, press **Enter**. If a default answer is not available (for example, the device hostname), the device uses what was previously configured and skips to the next question.

This figure shows how to enter and exit the setup script.

Figure 2: Setup Script Flow



You use the setup utility mainly for configuring the system initially, when no configuration is present. However, you can use the setup utility at any time for basic device configuration. The setup utility keeps the configured values when you skip steps in the script. For example, if you have already configured the mgmt0 interface, the setup utility does not change that configuration if you skip that step. However, if there is a default value for the step, the setup utility changes to the configuration using that default, not the configured value. Be sure to carefully check the configuration changes before you save the configuration.

Note Be sure to configure the IPv4 route, the default network IPv4 address, and the default gateway IPv4 address to enable SNMP access. If you enable IPv4 routing, the device uses the IPv4 route and the default network IPv4 address. If IPv4 routing is disabled, the device uses the default gateway IPv4 address.

Note The setup script only supports IPv4.

Prerequisites for the Setup Utility

The setup utility has the following prerequisites:

- Have a password strategy for your network environment.
- Connect the console port on the supervisor module to the network. If you have dual supervisor modules, connect the console ports on both supervisor modules to the network.
- Connect the Ethernet management port on the supervisor module to the network. If you have dual supervisor modules, connect the Ethernet management ports on both supervisor modules to the network.
- Enable the licensing grace period, if applicable. For detailed information about licensing, see the *Cisco NX-OS Licensing Guide*.

Setting Up Your Cisco NX-OS Device

To configure basic management of the Cisco NX-OS device using the setup utility, follow these steps:

Procedure

Step 1 Power on the device.

Step 2 Enable or disable password-strength checking.
A strong password has the following characteristics:

- At least eight characters long
- Does not contain many consecutive characters (such as "abcd")
- Does not contain many repeating characters (such as "aaabbb")
- Does not contain dictionary words
- Does not contain proper names
- Contains both uppercase and lowercase characters
- Contains numbers

Example:

```
         ---- System Admin Account Setup ----

Do you want to enforce secure password standard (yes/no) [y]: y
```

Step 3 Enter the new password for the administrator.

Note If a password is trivial (such as a short, easy-to-decipher password), your password configuration is rejected. Passwords are case sensitive. Be sure to configure a strong password that has at least eight characters, both uppercase and lowercase letters, and numbers.

Example:

```
Enter the password for "admin": <password>

Confirm the password for "admin": <password>

---- Basic System Configuration Dialog VDC: 1 ----

This setup utility will guide you through the basic configuration of
the system. Setup configures only enough connectivity for management
of the system.

Please register Cisco Nexus7000 Family devices promptly with your
supplier. Failure to register may affect response times for initial
service calls. Nexus7000 devices must be registered to receive
entitled support services.

Press Enter at anytime to skip a dialog. Use ctrl-c at anytime
to skip the remaining dialogs.
```

Step 4 Enter the setup mode by entering yes.

Example:

```
Would you like to enter the basic configuration dialog (yes/no): yes
```

Step 5 Create additional accounts by entering yes (no is the default).

Example:

```
  Create another login account (yes/no) [n]:yes
```

a) Enter the user login ID.

Example:

```
Enter the User login Id : user_login
```

Caution Usernames must begin with an alphanumeric character and can contain only these special characters: (+ = . _ \ -). The # and ! symbols are not supported. If the username contains characters that are not allowed, the specified user is unable to log in.

b) Enter the user password.

Example:

```
Enter the password for "user1": user_password
Confirm the password for "user1": user_password
```

c) Enter the default user role.

Example:

```
Enter the user role (network-operator|network-admin|vdc-operator|vdc-admin)
[network-operator]: default_user_role
```

For information on the default user roles, see the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 5.x*.

Step 6 Configure an SNMP community string by entering yes.

Example:

```
Configure read-only SNMP community string (yes/no) [n]: yes
SNMP community string : snmp_community_string
```

For information on SNMP, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 5.x*.

Step 7 Enter a name for the device (the default name is switch).

Example:

```
Enter the switch name: switch_name
```

Step 8 Enable the license grace period by entering yes.

Note Enabling the grace period allows users to test licensed features, except for Cisco TrustSec, which requires an Advanced Services license. The grace period is 120 days and starts when you first configure a licensed feature and stops when all features for a license are disabled. Once the grace period expires, you must purchase the license to access the licensed features. For more information about licenses, see the *Cisco NX-OS Licensing Guide*.

Example:

```
Enable license grace period? (yes/no) [n]: yes
```

Step 9 Configure out-of-band management by entering **yes**. You can then enter the mgmt0 IPv4 address and subnet mask.

Note You can only configure IPv4 address in the setup utility. For information on configuring IPv6, see the *Cisco Nexus 7000 Series NX-OS Unicast Routing Configuration Guide, Release 5.x*.

Example:

```
Continue with Out-of-band (mgmt0) management configuration? [yes/no]: yes
Mgmt0 IPv4 address: mgmt0_ip_address
Mgmt0 IPv4 netmask: mgmt0_subnet_mask
```

Step 10 Configure the IPv4 default gateway (recommended) by entering yes. You can then enter its IP address.

Example:

```
Configure the default-gateway: (yes/no) [y]: yes
IPv4 address of the default-gateway: default_gateway
```

Step 11 Configure advanced IP options such as the static routes, default network, DNS, and domain name by entering yes.

Example:

```
Configure Advanced IP options (yes/no)? [n]: yes
```

Step 12 Configure a static route (recommended) by entering yes. You can then enter its destination prefix, destination prefix mask, and next hop IP address.

Example:

```
Configure static route: (yes/no) [y]: yes
Destination prefix: dest_prefix
Destination prefix mask: dest_mask
Next hop ip address: next_hop_address
```

Step 13 Configure the default network (recommended) by entering yes. You can then enter its IPv4 address.

Note The default network IPv4 address is the same as the destination prefix in the static route configuration.

Example:

```
Configure the default network: (yes/no) [y]: yes
Default network IP address [dest_prefix]: dest_prefix
```

Step 14 Configure the DNS IPv4 address by entering yes. You can then enter the address.

Example:

```
Configure the DNS IP address? (yes/no) [y]: yes
DNS IP address: ipv4_address
```

Step 15 Configure the default domain name by entering yes. You can then enter the name.

Example:

```
Configure the DNS IP address? (yes/no) [y]: yes
DNS IP address: ipv4_address
```

Step 16 Enable the Telnet service by entering yes.

Example:

```
Enable the telnet service? (yes/no) [y]: yes
```

Step 17 Enable the SSH service by entering yes. You can then enter the key type and number of key bits. For more information, see the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 5.x*.

Example:

```
Enable the ssh service? (yes/no) [y]: yes
Type of ssh key you would like to generate (dsa/rsa) : key_type
Number of  key bits <768-2048> : number_of_bits
```

Step 18 Configure the NTP server by entering yes. You can then enter its IP address. For more information, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 5.x*.

Example:

```
Configure NTP server? (yes/no) [n]: yes
NTP server IP address: ntp_server_IP_address
```

Step 19 Specify a default interface layer (L2 or L3).

Example:

```
Configure default interface layer (L3/L2) [L3]: interface_layer
```

Step 20 Enter the default switchport interface state (shutdown or no shutdown). A shutdown interface is in an administratively down state. For more information, see the *Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide, Release 5.x*.

Example:

```
Configure default switchport interface state (shut/noshut) [shut]: default_state
```

Step 21 Enter the best practices profile for control plane policing (CoPP). For more information, see the *Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 5.x*.

Example:

```
Configure best practices CoPP profile (strict/moderate/lenient/none) [strict]: policy
```

Step 22 Configure CMP for the current supervisor, and then enter the IP address, netmask, and default gateway IP by entering yes. For more information, see the *Cisco Nexus 7000 Series Connectivity Management Processor Configuration Guide*.

Example:

```
Configure CMP processor on current sup (slot 5)? (yes/no) [y]: yes
cmp-mgmt IPv4 address : IP_address
cmp-mgmt IPv4 netmask : net_mask
IPv4 address of the default gateway : default_gateway
```

Step 23 Configure CMP for the redundant supervisor by entering yes. You can then enter the IP address, netmask, and default gateway IP.

Example:

```
Configure CMP processor on standby sup (slot 5)? (yes/no) [y]: yes
cmp-mgmt IPv4 address : IP_address
cmp-mgmt IPv4 netmask : net_mask
IPv4 address of the default gateway : default_gateway
```

The system now summarizes the complete configuration and asks if you want to edit it.

Step 24 Continue to the next step by entering no. If you enter yes, the setup utility returns to the beginning of the setup and repeats each step.

Example:

```
Would you like to edit the configuration? (yes/no) [y]: yes
```

Step 25 Use and save this configuration by entering yes. If you do not save the configuration at this point, none of your changes are part of the configuration the next time the device reboots. Enter yes to save the new configuration. This ensures that the boot variables for the kickstart and system images are also automatically configured.

Example:

```
Use this configuration and save it? (yes/no) [y]: yes
```

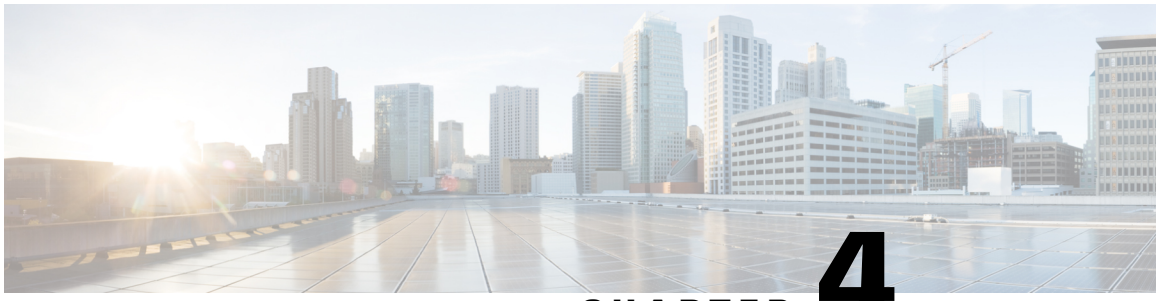
Caution If you do not save the configuration at this point, none of your changes are part of the configuration the next time that the device reboots. Enter yes to save the new configuration to ensure that the boot variables for the kickstart and system images are also automatically configured.

Additional References for the Setup Utility

This section includes additional information related to using the setup utility.

Related Documents for the Setup Utility

Related Topic	Document Title
Licensing	*Cisco NX-OS Licensing Guide*
Command reference	*Cisco Nexus 7000 Series NX-OS Fundamentals Command Reference*
CMP	*Cisco Nexus 7000 Series Connectivity Management Processor Configuration Guide*
SSH and Telnet	*Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 5.x*
User roles	*Cisco Nexus 7000 Series NX-OS Security Configuration Guide, Release 5.x*
IPv4 and IPv6	*Cisco Nexus 7000 Series NX-OS Interfaces Configuration Guide, Release 5.x*
SNMP and NTP	*Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 5.x*



CHAPTER 4

Understanding the Command-Line Interface

This chapter contains the following sections:

Information About the CLI Prompt

Once you have successfully accessed the device, the CLI prompt displays in the terminal window of your console port or remote workstation as shown in the following example:

```
User Access Verification
login: admin
Password:<password>
Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Copyright (c) 2002-2009, 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#
```

You can change the default device hostname.

From the CLI prompt, you can do the following:

- Use CLI commands for configuring features
- Access the command history
- Use command parsing functions

Note In normal operation, usernames are case sensitive. However, when you are connected to the device through its console port, you can enter a login username in all uppercase letters regardless of how the username was defined. As long as you provide the correct password, the device logs you in.

Command Modes

This section describes command modes in the Cisco NX-OS CLI.

EXEC Command Mode

When you first log in, the Cisco NX-OS software places you in EXEC mode. The commands available in EXEC mode include the **show** commands that display the device status and configuration information, the **clear** commands, and other commands that perform actions that you do not save in the device configuration.

Global Configuration Command Mode

Global configuration mode provides access to the broadest range of commands. The term indicates characteristics or features that affect the device as a whole. You can enter commands in global configuration

mode to configure your device globally, or to enter more specific configuration modes to configure specific elements such as interfaces or protocols.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.
		Note The CLI prompt changes to indicate that you are in global configuration mode.

Interface Configuration Command Mode

One example of a specific configuration mode that you enter from global configuration mode is interface configuration mode. To configure interfaces on your device, you must specify the interface and enter interface configuration mode.

You must enable many features on a per-interface basis. Interface configuration commands modify the operation of the interfaces on the device, such as Ethernet interfaces or management interfaces (mgmt 0).

For more information about configuring interfaces, see the Cisco Nexus Interfaces guide for your device.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface** *type number* **Example:** switch(config)# interface ethernet 2/2 switch(config-if)#	Specifies the interface that you want to configure. The CLI places you into interface configuration mode for the specified interface. **Note** The CLI prompt changes to indicate that you are in interface configuration mode.

Subinterface Configuration Command Mode

From global configuration mode, you can access a configuration submode for configuring VLAN interfaces called subinterfaces. In subinterface configuration mode, you can configure multiple virtual interfaces on a single physical interface. Subinterfaces appear to a protocol as distinct physical interfaces.

Subinterfaces also allow multiple encapsulations for a protocol on a single interface. For example, you can configure IEEE 802.1Q encapsulation to associate a subinterface with a VLAN.

For more information about configuring subinterfaces, see the Cisco Nexus Interfaces guide for your device.. For details about the subinterface commands, see the command reference guide for your device.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**interface** *type number.subint*
	
	Example:
	`switch(config)# interface ethernet 2/2.1`
	`switch(config-subif)#`
		Specifies the VLAN interface to be configured.
		The CLI places you into a subinterface configuration mode for the specified VLAN interface.
		Note The CLI prompt changes to indicate that you are in global configuration mode.

Saving and Restoring a Command Mode

The Cisco NX-OS software allows you to save current command mode, configure a feature, and then restore the previous command mode. The **push** command saves the command mode and the **pop** command restores the command mode.

The following example shows how to save and restore a command mode:

```
switch# configure terminal
switch(config)# event manager applet test
switch(config-applet)# push
switch(config-applet)# configure terminal
switch(config)# username testuser password newtest
switch(config)# pop
switch(config-applet)#
```

Exiting a Configuration Command Mode

To exit from any configuration command mode, perform one of the following tasks:

Procedure

	Command or Action	Purpose

Step 1	**exit**
	
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits from the current configuration command mode and returns to the previous configuration command mode.

Step 2	**end**
	
	Example:
	`switch(config-if)# end`
	`switch#`
		Exits from the current configuration command mode and returns to EXEC mode.

Step 3	**Ctrl-Z**
	
	Example:
	`switch(config-if)# ^z`
	`switch#`
		(Optional)
		Exits the current configuration command mode and returns to EXEC mode.
		Caution If you use **Ctrl-Z** at the end of a command line in which a valid command has been typed, the CLI adds the command to the running configuration file. In most cases, you should exit a configuration mode using the **exit** or **end** command.

Command Mode Summary

This table summarizes information about the main command modes.

Table 4: Command Mode Summary

Mode	Access Method	Prompt	Exit Method
EXEC	From the login prompt, enter your username and password.	`switch#`	To exit to the login prompt, use the **exit** command.
Global configuration	From EXEC mode, use the **configure terminal** command.	`switch(config)#`	To exit to EXEC mode, use the **end** or **exit** command or press **Ctrl-Z**.
Interface configuration	From global configuration mode, use an interface command and specify an interface with an **interface** command.	`switch(config-if)#`	To exit to global configuration mode, use the **exit** command. To exit to EXEC mode, use the **exit** command or press **Ctrl-Z**.
Subinterface configuration	From global configuration mode, specify a subinterface with an **interface** command.	`switch(config-subif)#`	To exit to global configuration mode, use the **exit** command. To exit to EXEC mode, use the **end** command or press **Ctrl-Z**.
VDC configuration	From global configuration mode, use the **vdc** command and specify a VDC name.	`switch(config-vdc)#`	To exit to global configuration mode, use the **exit** command. To exit to EXEC mode, use the **end** command or press **Ctrl-Z**.
VRF configuration	From global configuration mode, use the **vrf** command and specify a routing protocol.	`switch(config-vrf)#`	To exit to global configuration mode, use the **exit** command. To exit to EXEC mode, use the **end** command or press **Ctrl-Z**.
EXEC for a nondefault VDC	From EXEC mode, use the **switchto vdc** command and specify a VDC.	`switch-vdc2#`	To exit to the default VDC, use the **exit** command or the **switchback** command.
EXEC for a nondefault VRF		`switch%red#`	

Mode	Access Method	Prompt	Exit Method
	From EXEC mode, use the **routing-context vrf** command and specify a VRF.		To exit to the default VRF, use the **routing-context vrf default** command.

Special Characters

This table lists the characters that have special meaning in Cisco NX-OS text strings and should be used only in regular expressions or other special contexts.

Table 5: Special Characters

Character	Description
%	Percent
#	Pound, hash, or number
...	Ellipsis
|	Vertical bar
< >	Less than or greater than
[ ]	Brackets
{ }	Braces

Keystroke Shortcuts

This table lists command key combinations that can be used in both EXEC and configuration modes.

Table 6: Keystroke Shortcuts

Keystokes	Description
Ctrl-A	Moves the cursor to the beginning of the line.

Keystokes	Description
Ctrl-B	Moves the cursor one character to the left. When you enter a command that extends beyond a single line, you can press the Left Arrow or Ctrl-B keys repeatedly to scroll back toward the system prompt and verify the beginning of the command entry, or you can press the Ctrl-A key combination.
Ctrl-C	Cancels the command and returns to the command prompt.
Ctrl-D	Deletes the character at the cursor.
Ctrl-E	Moves the cursor to the end of the line.
Ctrl-F	Moves the cursor one character to the right.
Ctrl-G	Exits to the previous command mode without removing the command string.
Ctrl-K	Deletes all characters from the cursor to the end of the command line.
Ctrl-L	Redisplays the current command line.
Ctrl-N	Displays the next command in the command history.
Ctrl-O	Clears the terminal screen.
Ctrl-P	Displays the previous command in the command history.
Ctrl-R	Redisplays the current command line.
Ctrl-T	Transposes the character under the cursor with the character located to the right of the cursor. The cursor is then moved right one character.
Ctrl-U	Deletes all characters from the cursor to the beginning of the command line.
Ctrl-V	Removes any special meaning for the following keystroke. For example, press Ctrl-V before entering a question mark (?) in a regular expression.
Ctrl-W	Deletes the word to the left of the cursor.

Keystokes	Description
Ctrl-X, H	Lists the history of commands you have entered. When using this key combination, press and release the Ctrl and X keys together before pressing H.
Ctrl-Y	Recalls the most recent entry in the buffer (press keys simultaneously).
Ctrl-Z	Ends a configuration session, and returns you to EXEC mode. When used at the end of a command line in which a valid command has been typed, the resulting configuration is first added to the running configuration file.
Up arrow key	Displays the previous command in the command history.
Down arrow key	Displays the next command in the command history.
Right arrow key Left arrow key	Moves your cursor through the command string, either forward or backward, allowing you to edit the current command.
?	Displays a list of available commands.

Keystokes	Description

Tab	Completes the word for you after entering the first characters of the word, and then pressing the Tab key. All options that match are presented.
	Use tabs to complete the following items:
	• Command names
	• Scheme names in the file system
	• Server names in the file system
	• Filenames in the file system
	Example:
	`switch(config)# xm<Tab>`
	`switch(config)# xml<Tab>`
	`switch(config)# xml server`

	Example:
	`switch(config)# c<Tab>`
	`callhome  class-map  clock  cts`
	`cdp  cli  control-plane`
	`switch(config)# cl<Tab>`
	`class-map  cli  clock`
	`switch(config)# cla<Tab>`
	`switch(config)# class-map`

	Example:
	`switch# cd bootflash:<Tab>`
	`bootflash:  bootflash://sup-1/`
	`bootflash:///  bootflash://sup-2/`
	`bootflash://module-5/`
	`bootflash://sup-active/`
	`bootflash://module-6/  bootflash://sup-local/`

	Example:
	`switch# cd bootflash://mo<Tab>`
	`bootflash://module-5/`
	`bootflash://module-6/cv`
	`switch# cd bootflash://module-`

Abbreviating Commands

You can abbreviate commands and keywords by entering the first few characters of a command. The abbreviation must include sufficient characters to make it unique from other commands or keywords. If you are having trouble entering a command, check the system prompt and enter the question mark (?) for a list of available commands. You might be in the wrong command mode or using incorrect syntax.

This table lists examples of command abbreviations.

Table 7: Examples of Command Abbreviations

Command	Abbreviation
configure terminal	conf t
copy running-config startup-config	copy run start
interface ethernet 1/2	int e 1/2
show running-config	sh run

Completing a Partial Command Name

If you cannot remember a complete command name, or if you want to reduce the amount of typing you have to perform, enter the first few letters of the command, then press the **Tab** key. The command line parser will complete the command if the string entered is unique to the command mode. If your keyboard does not have a **Tab** key, press **Ctrl-I** instead.

The CLI recognizes a command once you have entered enough characters to make the command unique. For example, if you enter "conf" in EXEC mode, the CLI will be able to associate your entry with the **configure** command, because only the **configure** command begins with "conf".

In the following example the CLI recognizes the unique string for **conf** in EXEC mode when you press the **Tab** key:

```
switch# conf<Tab>
switch# configure
```

When you use the command completion feature the CLI displays the full command name. The CLI does not execute the command until you press the **Return** or **Enter** key. This allows you to modify the command if the full command was not what you intended by the abbreviation. If you enter a set of characters that could indicate more than one command, a list of matching commands displays.

For example, entering **co<Tab>** lists all commands available in EXEC mode beginning with "co":

```
switch# co<Tab>
configure   copy
switch# co
```

Note that the characters you entered appear at the prompt again to allow you to complete the command entry.

Identifying Your Location in the Command Hierarchy

Some features have a configuration submode hierarchy nested more than one level. In these cases, you can display information about your present working context (PWC).

Procedure

	Command or Action	Purpose

Step 1	**where detail**
	
	Example:
	`switch# configure terminal`
	`switch(config)# interface mgmt0`
	`switch(config-if)# where detail`
	`mode:  conf`
	`  interface mgmt0`
	`  username:  admin`
	`  vdc:  switch`
	`  routing-context vrf: default`
		Displays the PWC.

Using the no Form of a Command

Almost every configuration command has a **no** form that can be used to disable a feature, revert to a default value, or remove a configuration. The Cisco NX-OS command reference publications describe the function of the **no** form of the command whenever a **no** form is available.

This example shows how to disable a feature:

```
switch# configure terminal
switch(config)# feature tacacs+
switch(config)# no feature tacacs+
```

This example shows how to revert to the default value for a feature:

```
switch# configure terminal
switch(config)# banner motd #Welcome to the switch#
switch(config)# show banner motd
Welcome to the switch

switch(config)# no banner motd
switch(config)# show banner motd
User Access Verification
```

This example shows how to remove the configuration for a feature:

```
switch# configure terminal
switch(config)# radius-server host 10.10.2.2
switch(config)# show radius-server
retransmission count:0
timeout value:1
deadtime value:1
total number of servers:1

following RADIUS servers are configured:
        10.10.1.1:
                available for authentication on port:1812
                available for accounting on port:1813
        10.10.2.2:
                available for authentication on port:1812
                available for accounting on port:1813

switch(config)# no radius-server host 10.10.2.2
switch(config)# show radius-server
```

```
retransmission count:0
timeout value:1
deadtime value:1
total number of servers:1

following RADIUS servers are configured:
        10.10.1.1:
                available for authentication on port:1812
                available for accounting on port:1813
```

This example shows how to use the **no** form of a command in EXEC mode:

```
switch# cli var name testinterface ethernet1/2
switch# show cli variables
SWITCHNAME="switch"
TIMESTAMP="2009-05-12-13.43.13"
testinterface="ethernet1/2"

switch# cli no var name testinterface
switch# show cli variables
SWITCHNAME="switch"
TIMESTAMP="2009-05-12-13.43.13"
```

Configuring CLI Variables

This section describes CLI variables in the Cisco NX-OS CLI.

About CLI Variables

The Cisco NX-OS software supports the definition and use of variables in CLI commands.

You can refer to CLI variables in the following ways:

- Entered directly on the command line.
- Passed to a script initiated using the **run-script** command. The variables defined in the parent shell are available for use in the child **run-script** command process.

CLI variables have the following characteristics:

- Cannot have nested references through another variable
- Can persist across switch reloads or exist only for the current session

Cisco NX-OS supports one predefined variable: TIMESTAMP. This variable refers to the current time when the command executes in the format YYYY-MM-DD-HH.MM.SS.

Note The TIMESTAMP variable name is case sensitive. All letters must be uppercase.

Configuring CLI Session-Only Variables

You can define CLI session variables to persist only for the duration of your CLI session. These variables are useful for scripts that you execute periodically. You can reference the variable by enclosing the name in parentheses and preceding it with a dollar sign ($), for example $(*variable-name*).

Procedure

	Command or Action	Purpose
Step 1	**cli var name** *variable-name variable-text* **Example:** `switch# cli var name testinterface ethernet 2/1`	Configures the CLI session variable. The *variable-name* argument is alphanumeric, case sensitive, and has a maximum length of 31 characters. The *variable-text* argument is alphanumeric, case sensitive, can contain spaces, and has a maximum length of 200 characters.
Step 2	**show cli variables** **Example:** `switch# show cli variables`	(Optional) Displays the CLI variable configuration.

Configuring Persistent CLI Variables

You can configure CLI variables that persist across CLI sessions and device reloads.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**cli var name** *variable-name variable-text* **Example:** `switch(config)# cli var name testinterface ethernet 2/1`	Configures the CLI persistent variable. The variable name is case-sensitive alphanumeric string and must begin with an alphabetic character. The maximum length is 31 characters.

Step 3	**exit**
	
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	**show cli variables** **Example:** `switch# show cli variables`	(Optional) Displays the CLI variable configuration.
Step 5	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Command Aliases

This section provides information about command aliases.

About Command Aliases

You can define command aliases to replace frequently used commands. The command aliases can represent all or part of the command syntax.

Command alias support has the following characteristics:

- Command aliases are global for all user sessions.
- Command aliases persist across reboots if you save them to the startup configuration.
- Command alias translation always takes precedence over any keyword in any configuration mode or submode.
- Command alias configuration takes effect for other user sessions immediately.
- The Cisco NX-OS software provides one default alias, **alias**, which is the equivalent to the **show cli alias** command that displays all user-defined aliases.
- You cannot delete or change the default command alias **alias**.
- You can nest aliases to a maximum depth of 1. One command alias can refer to another command alias that must refer to a valid command, not to another command alias.
- A command alias always replaces the first command keyword on the command line.
- You can define command aliases for commands in any command mode.
- If you reference a CLI variable in a command alias, the current value of the variable appears in the alias, not the variable reference.
- You can use command aliases for **show** command searching and filtering.

Defining Command Aliases

You can define command aliases for commonly used commands.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose
Step 2	**cli alias name** *alias-name alias-text* **Example:** `switch(config)# cli alias name ethint interface ethernet`	Configures the command alias. The alias name is an alphanumeric string that is not case sensitive and must begin with an alphabetic character. The maximum length is 30 characters.

Step 3	**exit**
	
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	**alias** **Example:** `switch# alias`	(Optional) Displays the command alias configuration.
Step 5	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring Command Aliases for a User Session

You can create a command alias for the current user session which is not available to any other user on the Cisco NX-OS device. You can also save the command alias for future use by the current user account.

Procedure

	Command or Action	Purpose
Step 1	**terminal alias** [**persist**] *alias-name command -string* **Example:** `switch# terminal alias shintbr show interface brief`	Configures a command alias for the current user session. Use the **persist** keyword to save the alias for future use by the user account. **Note** Do not abbreviate the **persist** keyword.

Command Scripts

This section describes how you can create scripts of commands to perform multiple tasks.

Running a Command Script

You can create a list of commands in a file and execute them from the CLI. You can use CLI variables in the command script.

Note You cannot create the script files at the CLI prompt. You can create the script file on a remote device and copy it to the bootflash:, slot0:, or volatile: directory on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose
Step 1	**run-script** [**bootflash:** | **slot0:** | **volatile:**]*filename* **Example:** `switch# run-script testfile`	Executes the commands in the file on the default directory.

Echoing Information to the Terminal

You can echo information to the terminal, which is particularly useful from a command script. You can reference CLI variables and use formatting options in the echoed text.

This table lists the formatting options that you can insert in the text.

Table 8: Formatting Options for the echo Command

Formatting Option	Description
\b	Inserts back spaces.
\c	Removes the new line character at the end of the text string.
\f	Inserts a form feed character.
\n	Inserts a new line character.
\r	Returns to the beginning of the text line.
\t	Inserts a horizontal tab character.
\v	Inserts a vertical tab character.
\\	Displays a backslash character.

Formatting Option	Description
nnn	Displays the corresponding ASCII octal character.

Procedure

	Command or Action	Purpose

Step 1	**echo** [**backslash-interpret**] [*text*]
	
	Example:
	`switch# echo This is a test.`
	`This is a test.`
		The **backslash-interpret** keyword indicates that the text string contains formatting options. The *text* argument is alphanumeric, case sensitive, and can contain blanks. The maximum length is 200 characters. The default is a blank line.

Delaying Command Action

You can delay a command action for a period of time, which is particularly useful within a command script.

Procedure

	Command or Action	Purpose
Step 1	**sleep** *seconds* **Example:** `switch# sleep 30`	Causes a delay for a number of seconds. The range is from 0 to 2147483647.

Context-Sensitive Help

The Cisco NX-OS software provides context-sensitive help in the CLI. You can use a question mark (?) at any point in a command to list the valid input options.

CLI uses the caret (^) symbol to isolate input errors. The ^ symbol appears at the point in the command string where you have entered an incorrect command, keyword, or argument.

This table shows example outputs of context sensitive help.

Table 9: Context-Sensitive Help Example

Example Outputs	Description

`switch# clock ?`
`  set  HH:MM:SS Current Time`
`switch# clock`
	Displays the command syntax for the **clock** command in EXEC mode.
	The switch output shows that the **set** keyword is required for using the **clock** command.

`switch# clock set ?`
`  WORD  HH:MM:SS Current Time`
`switch# clock set`
	Displays the command syntax for setting the time.
	The help output shows that the current time is required for setting the clock and how to format the time.

`switch# clock set 13:32:00<CR>`
`% Incomplete command`
`switch#`
	Adds the current time.
	The CLI indicates the command is incomplete.

`switch# <Ctrl-P>` `switch# clock set 13:32:00`	Displays the previous command that you entered.

`switch# clock set 13:32:00 ?`
`  <1-31>  Day of the month`
`switch# clock set 13:32:00`
	Displays the additional arguments for the **clock set** command.

`switch# clock set 13:32:00 18 ?`
`  April  Month of the year`
`  August  Month of the year`
`  December  Month of the year`
`  February  Month of the year`
`  January  Month of the year`
`  July  Month of the year`
`  June  Month of the year`
`  March  Month of the year`
`  May  Month of the year`
`  November  Month of the year`
`  October  Month of the year`
`  September  Month of the year`
`switch# clock set 13:32:00 18`
	Displays the additional arguments for the **clock set** command.

`switch# clock set 13:32:00 18 April 08<CR>` `% Invalid input detected at '^' marker.`	Adds the date to the clock setting. The CLI indicates an error with the caret symbol (^) at 08.

`switch# clock set 13:32:00 18 April ?`
`  <2000-2030>  Enter the year (no abbreviation)`
`switch# clock set 13:32:00 18 April`
	Displays the correct arguments for the year.

`switch# clock set 13:32:00 18 April 2008<CR>` `switch#`	Enters the correct syntax for the **clock set** command.

Understanding Regular Expressions

The Cisco NX-OS software supports regular expressions for searching and filtering in CLI output, such as the **show** commands. Regular expressions are case sensitive and allow for complex matching requirements.

Special Characters

You can also use other keyboard characters (such as ! or ~) as single-character patterns, but certain keyboard characters have special meanings when used in regular expressions.

This table lists the keyboard characters that have special meanings.

Table 10: Special Characters with Special Meaning

Character	Special Meaning
.	Matches any single character, including white space.
*	Matches 0 or more sequences of the pattern.
+	Matches 1 or more sequences of the pattern.
?	Matches 0 or 1 occurrences of the pattern.
^	Matches the beginning of the string.
$	Matches the end of the string.
_ (underscore)	Matches a comma (,), left brace ({), right brace (}), left parenthesis (( ), right parenthesis ( )), the beginning of the string, the end of the string, or a space. **Note** The underscore is only treated as a regular expression for BPG related commands

To use these special characters as single-character patterns, remove the special meaning by preceding each character with a backslash (\). This example contains single-character patterns that match a dollar sign ($), an underscore (_), and a plus sign (+), respectively:

\$ _ \+

Multiple-Character Patterns

You can also specify a pattern that contains multiple characters by joining letters, digits, or keyboard characters that do not have special meanings. For example, a4% is a multiple-character regular expression.

With multiple-character patterns, the order is important. The regular expression a4% matches the character a followed by a 4 followed by a percent sign (%). If the string does not have a4%, in that order, pattern matching

fails. The multiple-character regular expression a. (the character a followed by a period) uses the special meaning of the period character to match the letter a followed by any single character. With this example, the strings ab, a!, or a2 are all valid matches for the regular expression.

You can remove the special meaning of a special character by inserting a backslash before it. For example, when the expression a\. is used in the command syntax, only the string a. will be matched.

Anchoring

You can match a regular expression pattern against the beginning or the end of the string by anchoring these regular expressions to a portion of the string using the special characters.

This table lists the special characters that you can use for anchoring.

Table 11: Special Characters Used for Anchoring

Character	Description
^	Matches the beginning of the string.
$	Matches the end of the string.

For example, the regular expression **^con** matches any string that starts with "con", and **sole$** matches any string that ends with "sole".

Note The ^ symbol can also be used to indicate the logical function "not" when used in a bracketed range. For example, the expression [^abcd] indicates a range that matches any single letter, as long as it is not a, b, c, or d.

Searching and Filtering show Command Output

Often, the output from **show** commands can be lengthy and cumbersome. The Cisco NX-OS software provides the means to search and filter the output so that you can easily locate information. The searching and filtering options follow a pipe character (|) at the end of the **show** command. You can display the options using the using the CLI context-sensitive help facility:

```
switch# show running-config | ?
  cut      Print selected parts of lines.
  diff     Show difference between current and previous invocation (creates temp files:
           remove them with 'diff-clean' command and don't use it on commands with big
           outputs, like 'show tech'!)
  egrep    Egrep - print lines matching a pattern
  grep     Grep - print lines matching a pattern
  head     Display first lines
  human    Output in human format
  last     Display last lines
  less     Filter for paging
  no-more  Turn-off pagination for command output
  perl     Use perl script to filter output
  section  Show lines that include the pattern as well as the subsequent lines that are
           more indented than matching line
  sed      Stream Editor
```

```
sort     Stream Sorter
sscp     Stream SCP (secure copy)
tr       Translate, squeeze, and/or delete characters
uniq     Discard all but one of successive identical lines
vsh      The shell that understands cli command
wc       Count words, lines, characters
xml      Output in xml format (according to .xsd definitions)
begin    Begin with the line that matches
count    Count number of lines
end      End with the line that matches
exclude  Exclude lines that match
include  Include lines that match
```

Filtering and Searching Keywords

The Cisco NX-OS CLI provides a set of keywords that you can use with the **show** commands to search and filter the command output.

This table lists the keywords for filtering and searching the CLI output.

Table 12: Filtering and Searching Keywords

Keyword Syntax	Description
begin *string* **Example:** `show version | begin Hardware`	Starts displaying at the line that contains the text that matches the search string. The search string is case sensitive.
count **Example:** `show running-config | count`	Displays the number of lines in the command output.
cut [**-d** *character*] {**-b** | **-c** | **-f** | **-s**} **Example:** `show file testoutput | cut -b 1-10`	Displays only the part of the output lines. You can display a number of bytes (**-b**), characters (**-vcut** [**-d** *character*] {**-b** | **-c** | **-f** | **-s**}), or fields (**-f**). You can also use the **-d** keyword to define a field delimiter other than the tag character default. The **-s** keyword suppress the display of line not containing the delimiter.
end *string* **Example:** `show running-config | end interface`	Displays all lines up to the last occurrence of the search string.
exclude *string* **Example:** `show interface brief | exclude down`	Displays all lines that do not include the search string. The search string is case sensitive.
head [**lines** *lines*] **Example:** `show logging logfile | head lines 50`	Displays the beginning of the output for the number of lines specified. The default number of lines is 10.

Keyword Syntax	Description
human **Example:** `show version | human`	Displays the output in normal format if you have previously set the output format to XML using the **terminal output xml** command.
include *string* **Example:** `show interface brief | include up`	Displays all lines that include the search string. The search string is case sensitive.
last [*lines*] **Example:** `show logging logfile | last 50`	Displays the end of the output for the number of lines specified. The default number of lines is 10.
no-more **Example:** `show interface brief | no-more`	Displays all the output without stopping at the end of the screen with the —More— prompt.
sscp *SSH-connection-name filename* **Example:** `show version | sscp MyConnection show_version_output`	Redirects the output using streaming secure copy (sscp) to a named SSH connection. You can create the SSH named connection using the **ssh name** command.
wc [**bytes** | **lines** | **words**] **Example:** `show file testoutput | wc bytes`	Displays counts of characters, lines, or words. The default is to display the number of lines, words, and characters.
xml **Example:** `show version | xml`	Displays the output in XML format.

diff Utility

You can compare the output from a **show** command with the output from the previous invocation of that command.

diff-clean [**all-session**] [**all-users**]

This table describes the keywords for the diff utility.

Keyword	Description
all-sessions	Removes diff temporary files from all sessions (past and present sessions) of the current user.
all-users	Removes diff temporary files from all sessions (past and present sessions) of all users.

The Cisco NX-OS software creates temporary files for the most current output for a **show** command for all current and previous users sessions. You can remove these temporary files using the **diff-clean** command.

diff-clean [**all-sessions** | **all-users**]

By default, the **diff-clean** command removes the temporary files for the current user's active session. The **all-sessions** keyword removes temporary files for all past and present sessions for the current user. The **all-users** keyword removes temporary files for all past and present sessions for the all users.

grep and egrep Utilities

You can use the Global Regular Expression Print (grep) and Extended grep (egrep) command-line utilities to filter the **show** command output.

The grep and egrep syntax is as follows:

{**grep** | **egrep**} [**count**] [**ignore-case**] [**invert-match**] [**line-exp**] [**line-number**] [**next** *lines*] [**prev** *lines*] [**word-exp**] *expression*}]

This table lists the **grep** and **egrep** parameters.

Table 13: grep and egrep Parameters

Parameter	Description
count	Displays only the total count of matched lines.
ignore-case	Specifies to ignore the case difference in matched lines.
invert-match	Displays lines that do not match the expression.
line-exp	Displays only lines that match a complete line.
line-number	Specifies to display the line number before each matched line.
next *lines*	Specifies the number of lines to display after a matched line. The default is 0. The range is from 1 to 999.
prev *lines*	Specifies the number of lines to display before a matched line. The default is 0. The range is from 1 to 999.
word-exp	Displays only lines that match a complete word.
expression	Specifies a regular expression for searching the output.

less Utility

You can use the less utility to display the contents of the **show** command output one screen at a time. You can enter less commands at the : prompt. To display all less commands you can use, enter h at the : prompt.

sed Utility

You can use the Stream Editor (sed) utility to filter and manipulate the **show** command output as follows:

sed *command*

The *command* argument contains sed utility commands.

sort Utility

You can use the sort utility to filter **show** command output.

The sort utility syntax is as follows:

sort [**-M**] [**-b**] [**-d**] [**-f**] [**-g**] [**-i**] [**-k** *field-number*[**.***char-position*][*ordering*]] [**-n**] [**-r**] [**-t** *delimiter*] [**-u**]
This table describes the sort utiliity parameters.

Table 14: sort Utility Parameters

Parameter	Description
-M	Sorts by month.
-b	Ignores leading blanks (space characters). The default sort includes the leading blanks.
-d	Sorts by comparing only blanks and alphanumeric characters. The default sort includes all characters.
-f	Folds lowercase characters into uppercase characters.
-g	Sorts by comparing a general numeric value.
-i	Sorts only using printable characters. The default sort includes nonprintable characters.
-k *field-number*[**.***char-position*][*ordering*]	Sorts according to a key value. There is no default key value.
-n	Sorts according to a numeric string value.
-r	Reverses order of the sort results. The default sort output is in ascending order.

Parameter	Description
-t *delimiter*	Sorts using a specified delimiter. The default delimiter is the space character.
-u	Removes duplicate lines from the sort results. The sort output displays the duplicate lines.

Searching and Filtering from the --More-- Prompt

You can search and filter output from `--More--` prompts in the **show** command output.

This table describes the `--More--` prompt commands.

Table 15: --More-- Prompt Commands

Commands	Description
[*lines*]<space>	Displays output lines for either the specified number of lines or the current screen size.
[*lines*]**z**	Displays output lines for either the specified number of lines or the current screen size. If you use the *lines* argument, that value becomes the new default screen size.
[*lines*]<return>	Displays output lines for either the specified number of lines or the current default number of lines. The initial default is 1 line. If you use the optional *lines* argument, that value becomes the new default number of lines to display for this command.
[*lines*]**d** or [*lines*]Ctrl+shift+D	Scrolls through output lines for either the specified number of lines or the current default number of lines. The initial default is 11 lines. If you use the optional *lines* argument, that value becomes the new default number of lines to display for this command.
q or **Q** or Ctrl-C	Exits the `--More--` prompt.
[*lines*]**s**	Skips forward in the output for either the specified number of lines or the current default number of lines and displays a screen of lines. The default is 1 line.
[*lines*]**f**	Skips forward in the output for either the specified number of screens or the current default number of screens and displays a screen of lines. The default is 1 screen.

Commands	Description
=	Displays the current line number.
[*count*]/*expression*	Skips to the line that matches the regular expression and displays a screen of output lines. Use the optional *count* argument to search for lines with multiple occurrences of the expression. This command sets the current regular expression that you can use in other commands.
[*count*]**n**	Skips to the next line that matches the current regular expression and displays a screen of output lines. Use the optional *count* argument to skip past matches.
{**!** | **:!**[*shell-cmd*]}	Executes the command specified in the *shell-cmd* argument in a subshell.
.	Repeats the previous command.

Using the Command History

The Cisco NX-OS software CLI allows you to access the command history for the current user session. You can recall and reissue commands, with or without modification. You can also clear the command history.

Recalling a Command

You can recall a command in the command history to optionally modify and enter again.

This example shows how to recall a command and reenter it:

```
switch(config)# show cli history
0  11:04:07   configure terminal
1  11:04:28   show interface ethernet 2/24
2  11:04:39     interface ethernet 2/24
3  11:05:13       no shutdown
4  11:05:19     exit
5  11:05:25   show cli history
switch(config)# !1
switch(config)# show interface ethernet 2/24
```

You can also use the **Ctrl-P** and **Ctrl-N** keystroke shortcuts to recall commands.

Controlling CLI History Recall

You can control the commands that you recall from the CLI history using the **Ctrl-P** and **Ctrl-N** keystroke shortcuts. By default, the Cisco NX-OS software recalls all commands from the current command mode and higher command modes. For example, if you are working in global configuration mode, the command recall keystroke shortcuts recall both EXEC mode and global configuration mode commands. Using the **terminal**

history no-exec-in-config command, you can avoid recalling EXEC mode commands when you are in a configuration mode.

Procedure

	Command or Action	Purpose
Step 1	[**no**] **terminal history no-exec-in-config** **Example:** `switch# terminal history no-exec-in-config`	Configures the CLI history to remove the EXEC commands when you use the recall keystroke shortcuts in a configuration mode. The default recalls EXEC commands. You can revert to the default using the **no** form of the command.

Configuring the CLI Edit Mode

You can recall commands from the CLI history using the **Ctrl-P** and **Ctrl-N** keystroke shortcuts and edit them before reissuing them. The default edit mode is emacs. You can change the edit mode to vi.

Procedure

	Command or Action	Purpose
Step 1	[**no**] **terminal edit-mode vi** [**persist**] **Example:** `switch# terminal edit-mode vi`	Changes the CLI edit mode to vi for the user session. The **persist** keyword makes the setting persistent across sessions for the current username. Use the **no** to revert to using emacs.

Displaying the Command History

You can display the command history using the **show cli history** command.

The **show cli history** command has the following syntax:

show cli history [*lines*] [**config-only** | **exec-only** | **this-mode-only**] [**unformatted**]

By default, the number of lines displayed is 12 and the output includes the command number and timestamp.

The example shows how to display default number of lines of the command history:

```
switch# show cli history
```

The example shows how to display 20 lines of the command history:

```
switch# show cli history 20
```

The example shows how to display only the configuration commands in the command history:

```
switch(config)# show cli history config-only
```

The example shows how to display only the EXEC commands in the command history:

```
switch(config)# show cli history exec-only
```

The example shows how to display only the commands in the command history for the current command mode:

```
switch(config-if)# show cli history this-mode-only
```

The example shows how to display only the commands in the command history without the command number and timestamp:

```
switch(config)# show cli history unformatted
```

Enabling or Disabling the CLI Confirmation Prompts

For many features, the Cisco NX-OS software displays prompts on the CLI that ask for confirmation before continuing. You can enable or disable these prompts. The default is enabled.

Procedure

	Command or Action	Purpose
Step 1	[**no**] **terminal dont-ask** [**persist**] **Example:** `switch# terminal dont-ask`	Disables the CLI confirmation prompt. The **persist** keyword makes the setting persistent across sessions for the current username. The default is enabled. Use the **no** form of the command to enable the CLI confirmation prompts.

Setting CLI Display Colors

You can change the CLI colors to display as follows:

- The prompt displays in green if the previous command succeeded.
- The prompt displays in red of the previous command failed.
- The user input displays in blue.
- The command output displays in the default color.

The default colors are those sent by the terminal emulator software.

Procedure

	Command or Action	Purpose
Step 1	**terminal color** [**evening**] [**persist**] **Example:** `switch# terminal color`	Sets the CLI display colors for the terminal session. The **evening** keyword is not supported. The **persist** keyword makes the setting persistent across sessions for the current username. The default setting is not persistent.

Sending Commands to Modules

You can send commands directly to modules from the supervisor module session using the **slot** command.

The **slot** has the following syntax:

slot *slot-number* [**quoted**] *command-string*

By default, the keyword and arguments in the *command-string* argument are space-separated. To send more than one command to a module, separate the commands with a space character, a semicolon character (;), and a space character.

The **quoted** keyword indicates that the command string begins and ends with double quotation marks ("). Use this keyword when you want to redirect the module command output to a filtering utility, such as diff, that is only supported on the supervisor module session.

The following example shows how to display and filter module information:

```
switch# slot 2 show version | grep lc
```

The following example shows how to filter module information on the supervisor module session:

```
switch# slot 2 quoted "show version" | diff
switch# slot 4 quoted "show version" | diff -c
*** /volatile/vsh_diff_1_root_8430_slot__quoted_show_version.old       Wed Apr 29 20:10:41
 2009
--- -   Wed Apr 29 20:10:41 2009
***************
*** 1,5 ****
! RAM 1036860 kB
! lc2
  Software
    BIOS:      version 1.10.6
    system:    version 4.2(1) [build 4.2(0.202)]
--- 1,5 ----
! RAM 516692 kB
! lc4
  Software
    BIOS:      version 1.10.6
    system:    version 4.2(1) [build 4.2(0.202)]
***************
*** 12,16 ****
  Hardware
      bootflash: 0 blocks (block size 512b)

!    uptime is 0 days 1 hours 45 minute(s) 34 second(s)

--- 12,16 ----
  Hardware
```

```
      bootflash: 0 blocks (block size 512b)

!    uptime is 0 days 1 hours 45 minute(s) 42 second(s)
```

BIOS Loader Prompt

When the supervisor modules power up, a specialized BIOS image automatically loads and tries to locate a valid kickstart image for booting the system. If a valid kickstart image is not found, the following BIOS loader prompt displays:

```
loader>
```

For information on how to load the Cisco NX-OS software from th `loader>` prompt, see the Cisco Nexus Troubleshooting guide for your device.

Examples Using the CLI

This section includes examples of using the CLI.

Defining Command Aliases

This example shows how to define command aliases:

```
cli alias name ethint interface ethernet
cli alias name shintbr show interface brief
cli alias name shintupbr shintbr | include up | include ethernet
```

This example shows how to use a command alias:

```
switch# configure terminal
switch(config)# ethint 2/3
switch(config-if)#
```

Using CLI Session Variables

You can reference a variable using the syntax **$(***variable-name***)**.
This example shows how to reference a user-defined CLI session variable:

```
switch# show interface $(testinterface)
Ethernet2/1 is down (Administratively down)
  Hardware is 10/100/1000 Ethernet, address is 0000.0000.0000 (bia 0019.076c.4dac)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA
  auto-duplex, auto-speed
  Beacon is turned off
  Auto-Negotiation is turned on
  Input flow-control is off, output flow-control is off
  Auto-mdix is turned on
  Switchport monitor is off
  Last clearing of "show interface" counters never
  5 minute input rate 0 bytes/sec, 0 packets/sec
```

```
  5 minute output rate 0 bytes/sec, 0 packets/sec
  L3 in Switched:
    ucast: 0 pkts, 0 bytes - mcast: 0 pkts, 0 bytes
  L3 out Switched:
    ucast: 0 pkts, 0 bytes - mcast: 0 pkts, 0 bytes
  Rx
    0 input packets 0 unicast packets 0 multicast packets
    0 broadcast packets 0 jumbo packets 0 storm suppression packets
    0 bytes
  Tx
    0 output packets 0 multicast packets
    0 broadcast packets 0 jumbo packets
    0 bytes
    0 input error 0 short frame 0 watchdog
    0 no buffer 0 runt 0 CRC 0 ecc
    0 overrun  0 underrun 0 ignored 0 bad etype drop
    0 bad proto drop 0 if down drop 0 input with dribble
    0 input discard
    0 output error 0 collision 0 deferred
    0 late collision 0 lost carrier 0 no carrier
    0 babble
    0 Rx pause 0 Tx pause 0 reset
```

Using the System-Defined Timestamp Variable

This example uses $(TIMESTAMP) when redirecting **show** command output to a file:

```
switch# show running-config > rcfg.$(TIMESTAMP)
Preparing to copy....done
switch# dir
       12667     May 01 12:27:59 2008  rcfg.2008-05-01-12.27.59

Usage for bootflash://sup-local
8192 bytes used
20963328 bytes free
20971520 bytes total
```

Running a Command Script

This example displays the CLI commands specified in the script file:

```
switch# show file testfile
configure terminal
interface ethernet 2/1
no shutdown
end
show interface ethernet 2/1
```

This example displays the **run-script** command execution output:

```
switch# run-script testfile
`configure terminal`
`interface ethernet 2/1`
`no shutdown`
`end`
`show interface ethernet 2/1 `
Ethernet2/1 is down (Link not connected)
  Hardware is 10/100/1000 Ethernet, address is 0019.076c.4dac (bia 0019.076c.4dac)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA
  Port mode is trunk
  auto-duplex, auto-speed
```

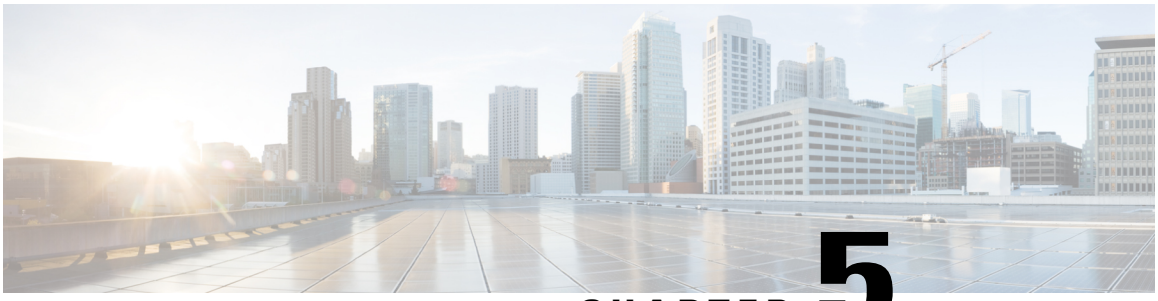
```
Beacon is turned off
Auto-Negotiation is turned on
Input flow-control is off, output flow-control is off
Auto-mdix is turned on
Switchport monitor is off
Last clearing of "show interface" counters 1d26.2uh
5 minute input rate 0 bytes/sec, 0 packets/sec
5 minute output rate 0 bytes/sec, 0 packets/sec
Rx
  0 input packets 0 unicast packets 0 multicast packets
  0 broadcast packets 0 jumbo packets 0 storm suppression packets
  0 bytes
Tx
  0 output packets 0 multicast packets
  0 broadcast packets 0 jumbo packets
  0 bytes
  0 input error 0 short frame 0 watchdog
  0 no buffer 0 runt 0 CRC 0 ecc
  0 overrun  0 underrun 0 ignored 0 bad etype drop
  0 bad proto drop 0 if down drop 0 input with dribble
  0 input discard
  0 output error 0 collision 0 deferred
  0 late collision 0 lost carrier 0 no carrier
  0 babble
  0 Rx pause 0 Tx pause 0 reset
```

Additional References for the CLI

This section includes additional information related to the CLI.

Related Documents for the CLI

Related Topic	Document Title
Cisco NX-OS Licensing	*Cisco NX-OS Licensing Guide*
Command reference	*Cisco Nexus 7000 Series NX-OS Fundamentals Command Reference*



CHAPTER 5

Configuring Terminal Settings and Sessions

This chapter contains the following sections:

Information About Terminal Settings and Sessions

This section includes information about terminal settings and sessions.

Terminal Session Settings

The Cisco NX-OS software features allow you to manage the following characteristics of terminals:

Terminal type

Name used by Telnet when communicating with remote hosts

Length

Number of lines of command output displayed before pausing

Width

Number of characters displayed before wrapping the line

Inactive session timeout

Number of minutes that a session remains inactive before the device terminates it

Console Port

The console port is an asynchronous serial port that allows you to connect to the device for initial configuration through a standard RS-232 port with an RJ-45 connector. Any device connected to this port must be capable of asynchronous transmission. You can configure the following parameters for the console port:

Data bits

Specifies the number of bits in an 8-bit byte that is used for data.

Inactive session timeout

Specifies the number of minutes a session can be inactive before it is terminated.

Parity

Specifies the odd or even parity for error detection.

Speed

Specifies the transmission speed for the connection.

Stop bits

Specifies the stop bits for an asynchronous line.

Configure your terminal emulator with 9600 baud, 8 data bits, 1 stop bit, and no parity.

COM1 Port

A COM1 port is an RS-232 port with a DB-9 interface that enables you to connect to an external serial communication device such as a modem. You can configure the following parameters for the COM1 port:

Data bits

Specifies the number of bits in an 8-bit byte that is used for data.

Hardware flowcontrol

Enables the flow-control hardware.

Parity

Specifies the odd or even parity for error detection.

Speed

Specifies the transmission speed for the connection.

Stop bits

Specifies the stop bits for an asynchronous line.

Configure your terminal emulator with 9600 baud, 8 data bits, 1 stop bit, and no parity.

Virtual Terminals

You can use virtual terminal lines to connect to your Cisco NX-OS device. Secure Shell (SSH) and Telnet create virtual terminal sessions. You can configure an inactive session timeout and a maximum sessions limit for virtual terminals.

Modem Support

You can connect a modem to the COM1 or console ports on the supervisor module. The following modems were tested on devices running the Cisco NX-OS software:

- MultiTech MT2834BA (http://www.multitech.com/en_us/support/families/multimodemii/)
- Hayes Accura V.92 (http://www.zoom.com/products/dial_up_external_serial.html#hayes)

Note Do not connect a modem when the device is booting. Only connect the modem when the device is powered-up.

The Cisco NX-OS software has the default initialization string (ATE0Q1&D2&C1S0=1\015) to detect connected modems. The default string is defined as follows:

AT

Attention

E0 (required)

No echo

Q1

Result code on

&D2

Normal data terminal ready (DTR) option

&C1

Enable tracking the state of the data carrier

S0=1

Pick up after one ring

\015 (required)

Carriage return in octal

Virtualization Support for Configuration Files

Except for removing the configuration for a missing module, the configuration file operations are local to the virtual device context (VDC). You can remove the missing module configuration only from the default VDC. For more information on VDCs, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide*.

Licensing Requirements for Terminal Settings and Sessions

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	Terminal setting configuration requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing scheme, see the *Cisco NX-OS Licensing Guide*.

Configuring the Console Port

You can set the following characteristics for the console port:

- Data bits
- Inactive session timeout
- Parity
- Speed
- Stop bits

Before You Begin

Log in to the console port.

Ensure that you are in the default VDC.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

	Command or Action	Purpose

Step 2	**line console**
	
	Example:
	`switch# line console`
	`switch(config-console)#`
		Enters console configuration mode.

Step 3	**databits** *bits* **Example:** `switch(config-console)# databits 7`	Configures the number of data bits per byte. The range is from 5 to 8. The default is 8.
Step 4	**exec-timeout** *minutes* **Example:** `switch(config-console)# exec-timeout 30`	Configures the timeout for an inactive session. The range is from 0 to 525600 minutes (8760 hours). A value of 0 minutes disables the session timeout. The default is 30 minutes.
Step 5	**parity** {**even** | **none** | **odd**} **Example:** `switch(config-console)# parity even`	Configures the parity. The default is **none**.
Step 6	**speed** {**300** | **1200** | **2400** | **4800** | **9600** | **38400** | **57600** | **115200**} **Example:** `switch(config-console)# speed 115200`	Configures the transmit and receive speed. The default is **115200**.
Step 7	**stopbits** {**1** | **2**} **Example:** `switch(config-console)# stopbits 2`	Configures the stop bits. The default is **1**.

Step 8	**exit**
	
	Example:
	`switch(config-console)# exit`
	`switch(config)#`
		Exits console configuration mode.

Step 9	**show line console** **Example:** `switch(config)# show line console`	(Optional) Displays the console settings.
Step 10	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring the COM1 Port

You can set the following characteristics for the COM1 port:

- Data bits
- Flow control on the hardware
- Parity
- Speed
- Stop bits

Before You Begin

Log in to the console port or COM1 port.

Ensure that you are in the default VDC.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**line com1**
	
	Example:
	`switch# line com1`
	`switch(config-com1)#`
		Enters COM1 configuration mode.

Step 3	**databits** *bits* **Example:** `switch(config-com1)# databits 7`	Configures the number of data bits per byte. The range is from 5 to 8. The default is 8.
Step 4	**flowcontrol hardware** **Example:** `switch(config-com1)# flowcontrol hardware`	Enables flow control on the hardware. The default is enabled. Use the **no flowcontrol hardware** command to disable flow control on the hardware.
Step 5	**parity** {**even** | **none** | **odd**} **Example:** `switch(config-com1)# parity even`	Configures the parity. The default is **none**.
Step 6	**speed** {**300** | **1200** | **2400** | **4800** | **9600** | **38400** | **57600** | **115200**} **Example:** `switch(config-com1)# speed 115200`	Configures the transmit and receive speed. The default is **9600**.

	Command or Action	Purpose
Step 7	**stopbits** {**1** | **2**} **Example:** `switch(config-com1)# stopbits 2`	Configures the stop bits. The default is **1**.

Step 8	**exit**
	
	Example:
	`switch(config-com1)# exit`
	`switch(config)#`
		Exits COM1 configuration mode.

Step 9	**show line com1** **Example:** `switch(config)# show line com1`	(Optional) Displays the COM1 port settings.
Step 10	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring Virtual Terminals

This section describes how to configure virtual terminals on Cisco NX-OS devices.

Configuring the Inactive Session Timeout

You can configure a timeout for inactive virtual terminal sessions on a Cisco NX-OS device.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**line vty**
	
	Example:
	`switch# line vty`
	`switch(config-line)#`
		Enters line configuration mode.

	Command or Action	Purpose
Step 3	**exec-timeout** *minutes* **Example:** `switch(config-line)# exec-timeout 30`	Configures the inactive session timeout for the VDC. The range is from 0 to 525600 minutes (8760 hours). A value of 0 minutes disables the timeout. The default value is 30.

Step 4	**exit**
	
	Example:
	`switch(config-line)# exit`
	`switch(config)#`
		Exits line configuration mode.

Step 5	**show running-config all | begin vty** **Example:** `switch(config)# show running-config all	begin vty`	(Optional) Displays the virtual terminal configuration.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring the Session Limit

You can limit the number of virtual terminal sessions on your Cisco NX-OS device.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**line vty**
	
	Example:
	`switch# line vty`
	`switch(config-line)#`
		Enters line configuration mode.

Step 3	**session-limit** *sessions* **Example:** `switch(config-line)# session-limit 10`	Configures the maximum number of virtual sessions for the Cisco NX-OS device. The range is from 1 to 64. The default is 32.

	Command or Action	Purpose

Step 4	**exit**
	
	Example:
	`switch(config-line)# exit`
	`switch(config)#`
		Exits line configuration mode.

Step 5	**show running-config all | being vty** **Example:** `switch(config)# show running-config all | begin vty`	(Optional) Displays the virtual terminal configuration.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring Modem Connections

You can connect a modem to either the COM1 port or the console port.

We recommend that you use the COM1 port to connect the modem.

Enabling a Modem Connection

You must enable the modem connection on the port before you can use the modem.

Before You Begin

Log in to the console port.

Ensure that you are in the default VDC.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands: **Command** / **Purpose** **line com1** — Enters COM1 configuration mode.	Enters COM1 configuration mode or console configuration mode.

	Command or Action	Purpose

	Command / **Purpose**: **line console** — Enters console configuration mode.
	
	Example:
	`switch# line com1`
	`switch(config-com1)#`	

Step 3	**modem in** **Example:** `switch(config-com1)# modem in`	Enables modem input on the COM1 or console port.

Step 4	**exit**
	
	Example:
	`switch(config-com1)# exit`
	`switch(config)#`
		Exits COM1 or console configuration mode.

Step 5	**show line** **Example:** `switch(config)# show line`	(Optional) Displays the console and COM1 settings.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Downloading the Default Initialization String

The Cisco NX-OS software provides a default initialization string that you can download for connecting with the modem. The default initialization string is ATE0Q1&D2&C1S0=1\015.

Before You Begin

Log in to the console port.

Ensure that you are in the default VDC.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands:
	Option / **Description**
	line com1 — Enters COM1 configuration mode.
	line console — Enters console configuration mode.
	
	Example:
	`switch# line com1`
	`switch(config-com1)#`	

Step 3	**modem init-string default** **Example:** `switch(config-com1)# modem init-string default`	Writes the default initialization string to the modem.

Step 4	**exit**
	
	Example:
	`switch(config-com1)# exit`
	`switch(config)#`
		Exits COM1 or console configuration mode.

Step 5	**show line** **Example:** `switch(config)# show line`	(Optional) Displays the COM1 and console settings.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring and Downloading a User-Specified Initialization String

You can configure and download your own initialization when the default initialization string is not compatible with your modem.

Before You Begin

Log in to the console port.

Ensure that you are in the default VDC.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	Enter one of the following commands:
	Option / **Description**
	line com1 — Enters COM1 configuration mode.
	line console — Enters console configuration mode.
	
	Example:
	`switch# line com1`
	`switch(config-com1)#`	

Step 3	**modem set-string user-input** *string* **Example:** `switch(config-com1)# modem set-string user-input ATE0Q1&D2&C1S0=3\015`	Sets the user-specified initialization string for the COM1 or console port. The initialization string is alphanumeric and case sensitive, can contain special characters, and has a maximum of 100 characters. **Note** You must first set the user-input string before initializing the string.
Step 4	**modem init-string user-input** **Example:** `switch(config-com1)# modem init-string user-input`	Writes the user-specified initialization string to the modem connected to the COM1 or console port.

Step 5	**exit**
	
	Example:
	`switch(config-com1)# exit`
	`switch(config)#`
		Exits COM1 or console configuration mode.

Step 6	**show line** **Example:** `switch(config)# show line`	(Optional) Displays the COM1 and console settings.

	Command or Action	Purpose
Step 7	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Initializing a Modem for a Powered-Up Cisco NX-OS Device

If you connect a modem to a powered-up physical device, you must initialize the modem before you can use it.

Before You Begin

After waiting until the Cisco NX-OS device has completed the boot sequence and the system image is running, connect the modem to either the COM1 port or the console port on the device.

Enable the modem connection on the port.

Ensure that you are in the default VDC.

Procedure

	Command or Action	Purpose
Step 1	**modem connect line** {**com1** | **console**} **Example:** `switch# modem connect line com1`	Initializes the modem connected to the device.

Related Topics

Enabling a Modem Connection, on page 69

Clearing Terminal Sessions

You can clear terminal sessions on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose
Step 1	**show users** **Example:** `switch# show users`	(Optional) Displays the user sessions on the device.

	Command or Action	Purpose
Step 2	**clear line** *name* **Example:** `switch# clear line pts/0`	Clears a terminal session on a specific line. The line name is case sensitive.

Displaying Terminal and Session Information

To display terminal and session information, perform one of the following tasks:

Command	Purpose
show terminal	Displays terminal settings.
show line	Displays the COM1 and console ports settings.
show users	Displays virtual terminal sessions.
show running-config [**all**]	Displays the user account configuration in the running configuration. The **all** keyword displays the default values for the user accounts.

For detailed information about the fields in the output from these commands, see the Cisco Nexus Command Reference guide for your device.

Default Settings for File System Parameters

This table lists the default settings for the file system parameters.

Table 16: Default File System Settings

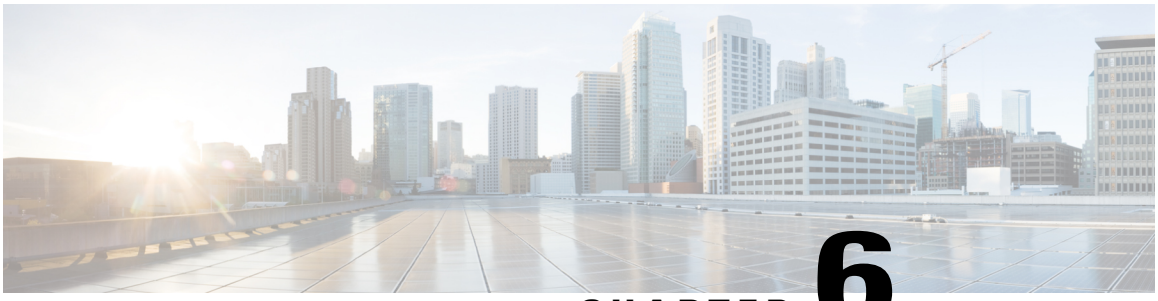
Parameters	Default
Default filesystem	bootflash:

Additional References for Terminal Settings and Sessions

This section includes additional references for terminal settings and sessions on NX-OS devices.

Related Documents for Terminal Settings and Sessions

Related Topic	Document Title
Licensing	*Cisco NX-OS Licensing Guide*
Command reference	*Cisco Nexus 7000 Series NX-OS Fundamentals Command Reference*



CHAPTER 6

Basic Device Management

This chapter contains the following sections:

Information About Basic Device Management

This section provides information about basic device management.

Device Hostname

You can change the device hostname displayed in the command prompt from the default (switch) to another character string. When you give the device a unique hostname, you can easily identify the device from the command-line interface (CLI) prompt.

Message-of-the-Day Banner

The message-of-the-day (MOTD) banner displays before the user login prompt on the device. This message can contain any information that you want to display for users of the device.

Device Clock

If you do not synchronize your device with a valid outside timing mechanism, such as an NTP clock source, you can manually set the clock time when your device boots. For information about NTP, see the *Cisco Nexus 7000 Series NX-OS System Management Configuration Guide, Release 5.x*.

Clock Manager

The Cisco Nexus chassis may contain clocks of different types that may need to be synchronized. These clocks are a part of various components (such as the supervisor, LC processors, or linecards) and each may be using a different protocol.

The clock manager provides a way to synchronize these different clocks.

Time Zone and Summer Time (Daylight Saving Time)

You can configure the time zone and summer time (daylight saving time) setting for your device. These values offset the clock time from Coordinated Universal Time (UTC). UTC is International Atomic Time (TAI) with leap seconds added periodically to compensate for the Earth's slowing rotation. UTC was formerly called Greenwich Mean Time (GMT).

User Sessions

You can display the active user session on your device. You can also send messages to the user sessions. For more information about managing user sessions and accounts, see the Cisco Nexus Security Configuration guide for your device.

Virtualization Support for Basic Device Management

Basic device management is local to the virtual device context (VDC). For more information on VDCs, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide*.

Licensing Requirements for Basic Device Management

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	Basic device management requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing scheme, see the *Cisco NX-OS Licensing Guide*.

Changing the Device Hostname

You can change the device hostname displayed in the command prompt from the default (switch) to another character string.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	{**hostname** | **switchname**} *name*
	
	Example:
	Using the **hostname** command:
	`switch(config)# hostname Engineering1`
	`Engineering1(config)#`
	Using the **switchname** command:
	`Engineering1(config)# switchname Engineering2`
	`Engineering2(config)#`
		Changes the device hostname. The *name* argument is alphanumeric, case sensitive, and has a maximum length of 32 characters. The default is switch.
		Note The **switchname** command performs the same function as the **hostname** command.

Step 3	**exit**
	
	Example:
	`Engineering2(config)# exit`
	`Engineering2#`
		Exits global configuration mode.

Step 4	**copy running-config startup-config** **Example:** `Engineering2# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring the MOTD Banner

You can configure the MOTD to display before the login prompt on the terminal when a user logs in. The MOTD banner has the following characteristics:

- Maximum of 80 characters per line
- Maximum of 40 lines

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**banner motd** *delimiting-character message delimiting-character*
	
	Example:
	`switch(config)# banner motd #Welcome to the Switch#`
	`switch(config)#`
		Configures the MOTD banner. Do not use the *delimiting-character* in the *message* text.
		Note Do not use " or % as a delimiting character.

Step 3	**exit**
	
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	**show banner motd** **Example:** `switch# show banner motd`	(Optional) Displays the configured MOTD banner.
Step 5	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring the Time Zone

You can configure the time zone to offset the device clock time from UTC.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**clock timezone** *zone-name offset-hours offset-minutes* **Example:** `switch(config)# clock timezone EST -5 0`	Configures the time zone. The *zone-name* argument is a 3-character string for the time zone acronym (for example, PST or EST). The *offset-hours* argument is the offset from the UTC and the range is from –23 to 23 hours. The range for the *offset-minutes* argument is from 0 to 59 minutes.

Step 3	**exit**
	
	Example:
	`switch(config)# exit`
	`switch#`
		Exits global configuration mode.

Step 4	**show clock** **Example:** `switch# show clock`	(Optional) Displays the time and time zone.
Step 5	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Configuring Summer Time (Daylight Saving Time)

You can configure when summer time, or daylight saving time, is in effect for the device and the offset in minutes.

Procedure

	Command or Action	Purpose

Step 1	**configure terminal**
	
	Example:
	`switch# configure terminal`
	`switch(config)#`
		Enters global configuration mode.

Step 2	**clock summer-time** *zone-name start-week start-day start-month start-time*	Configures summer time or daylight saving time.

	Command or Action	Purpose
	end-week end-day end-month end-time offset-minutes **Example:** `switch(config)# clock summer-time PDT 1 Sunday March 02:00 1 Sunday November 02:00 60`	The *zone-name* argument is a three character string for the time zone acronym (for example, PST and EST). The values for the *start-day* and *end-day* arguments are **Monday**, **Tuesday**, **Wednesday**, **Thursday**, **Friday**, **Saturday**, and **Sunday**. The values for the *start-month* and *end-month* arguments are **January**, **February**, **March**, **April**, **May**, **June**, **July**, **August**, **September**, **October**, **November**, and **December**. The value for the *start-time* and *end-time* arguments are in the format *hh*:*mm*. The range for the *offset-minutes* argument is from 0 to 1440 minutes.
Step 3	**exit** **Example:** `switch(config)# exit switch#`	Exits global configuration mode.
Step 4	**show clock detail** **Example:** `switch(config)# show clock detail`	(Optional) Displays the configured MOTD banner.
Step 5	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Manually Setting the Device Clock

You can set the clock manually if your device cannot access a remote time source.

Before You Begin

Configure the time zone.

Procedure

	Command or Action	Purpose
Step 1	**clock set** *time day month year*	Configures the device clock.

	Command or Action	Purpose

	Example:
	`switch# clock set 15:00:00 30 May 2008`
	`Fri May 30 15:14:00 PDT 2008`
		The format for the *time* argument is *hh*:*mm*:*ss*.
		The range for the *day* argument is from 1 to 31.
		The values for the *month* argument are **January**, **February**, **March**, **April**, **May**, **June**, **July**, **August**, **September**, **October**, **November**, and **December**.
		The range for the *year* argument is from 2000 to 2030.

Step 2	**show clock** **Example:** `switch(config)# show clock`	(Optional) Displays the current clock value.

Related Topics

Configuring the Time Zone, on page 80

Setting the Clock Manager

You can configure the clock manager to synchronize all the clocks of the components in the Cisco Nexus chassis.

Procedure

	Command or Action	Purpose
Step 1	**clock protocol** *protocol* **vdc** *vdc-num* **Example:** `#clock protocol ptp vdc 2`	Configures the clock manager. The values for the *protocol* argument are **ptp**, **ntp**, and **none**. The following describes the values: • **ptp**: Synchronize clocks with Precision Time Protocol (PTP) as described by IEEE 1588. • **ntp**: Synchronize clocks with Network Time Protocol (NTP). • **none**: Use 'clock set <HH:MM:SS>' to set SUP clocks. **Note** When **none** is used, the clock in the specified VDC must be configured. **Note** Once the protocol is configured, the clock in the specified VDC must use that protocol. For example, if the command 'clock protocol ptp vdc 2' is entered, then PTP should be configured in VDC 2. The range for the *vdc* argument is 1 to 8.

	Command or Action	Purpose
Step 2	**show run clock_manager** **Example:** `#show run clock_manager`	(Optional) Displays the configuration of the clock manager.

Managing Users

You can display information about users logged into the device and send messages to those users.

Displaying Information about the User Sessions

You can display information about the user session on the device.

Procedure

	Command or Action	Purpose
Step 1	**show users** **Example:** `switch# show users`	Displays the user sessions.

Sending a Message to Users

You can send a message to active users currently using the device CLI.

Procedure

	Command or Action	Purpose
Step 1	**show users** **Example:** `switch# show users`	(Optional) Displays the active user sessions.
Step 2	**send** [**session** *line*] *message-text* **Example:** `switch# send Reloading the device is 10 minutes!`	Sends a message to all active users or to a specific user. The message can be up to 80 alphanumeric characters and is case sensitive.

Verifying the Device Configuration

To verify the configuration after bootstrapping the device using POAP, use one of the following commands:

Command	Purpose
show running-config	Displays the running configuration.
show startup-config	Displays the startup configuration.

For detailed information about the fields in the output from these commands, see the Cisco Nexus Command Reference for your device.

Default Settings for Basic Device Parameters

This table lists the default settings for basic device parameters.

Table 17: Default Basic Device Parameters

Parameters	Default
MOTD banner text	User Access Verification
Clock time zone	UTC

Additional References for Basic Device Management

You can find additional information related to basic device management.

Related Documents for Basic Device Management

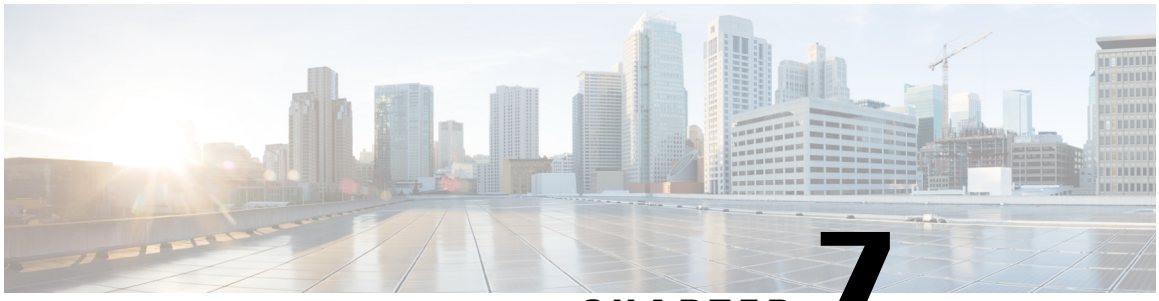
Related Topic	Document Title
Licensing	*Cisco NX-OS Licensing Guide*
Command reference	*Cisco Nexus 7000 Series NX-OS Fundamentals Command Reference*

Feature History for Basic Device Management

This table lists the release history for this feature.

Table 18: Feature History for File Systems, Directories, and Files

Feature Name	Releases	Feature Information	
Device management	4.0(1)	This feature was introduced.	



CHAPTER 7

Using the Device File Systems, Directories, and Files

This chapter contains the following sections:

Information About the Device File Systems, Directories, and Files

This section describes file systems, directories, and files on the Cisco NX-OS device.

File Systems

The syntax for specifying a local file system is *filesystem*:[//*modules*/].
This table describes file systems that you can reference on your device.

Table 19: File System Syntax Components

File System Name	Module	Description
bootflash	sup-active sup-local	Internal CompactFlash memory located on the active supervisor module used for storing image files, configuration files, and other miscellaneous files. The initial default directory is bootflash.
	sup-standby sup-remote	Internal CompactFlash memory located on the standby supervisor module used for storing image files, configuration files, and other miscellaneous files.
slot0	—	External CompactFlash memory installed in a supervisor module used for storing system images, configuration files, and other miscellaneous files.
volatile	—	Volatile random-access memory (VRAM) located on a supervisor module used for temporary or pending changes.
nvram	—	Nonvolatile random-access memory (NVRAM) located on a supervisor module used for storing the startup-configuration file.
log	—	Memory on the active supervisor that stores logging file statistics.
system	—	Memory on a supervisor module used for storing the running-configuration file.
debug	—	Memory on a supervisor module used for debug logs.
usb1	—	External USB flash memory installed in a supervisor module used for storing image files, configuration files, and other miscellaneous files.

File System Name	Module	Description
usb2	—	External USB flash memory installed in a supervisor module used for storing image files, configuration files, and other miscellaneous files.

Directories

You can create directories on bootflash: and external flash memory (slot0:, usb1:, and usb2:). You can navigate through these directories and use them for files.

Files

You create and access files on bootflash:. volatile:, slot0:, usb1:, and usb2: file systems. You can only access files on the system: file systems. You can use the debug: file system for debug log files specified in the **debug logfile** command.

You can download files, such as system image files, from remote servers using FTP, Secure Copy (SCP), Secure Shell FTP (SFTP), and TFTP. You can also copy files from an external server to the device, because the device can act as an SCP server.

Virtualization Support for File Systems

Most file system, directory, and file configuration and operations are local to the virtual device context (VDC). One exception is formatting an external Flash device, which you must perform from the default VDC. For more information on VDCs, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide*.

Licensing Requirements for File Systems, Directories, and Files

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	Using the file systems, directories, and files requires no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing scheme, see the *Cisco NX-OS Licensing Guide*.

Formatting External Flash Devices

You can format an external flash device to erase the contents from the default VDC and restore it to its factory-shipped state.

Note For information on recovering corrupted bootflash using formatting, see the Cisco Nexus 7000 Series NX-OS Troubleshooting Guide.

Before You Begin

Ensure you are in the default VDC.

Insert the external flash device in the active supervisor module.

Procedure

	Command or Action	Purpose
Step 1	**dir** {**slot0:** | **usb1:** | **usb2:**} **Example:** `switch# dir slot0:`	(Optional) Displays the contents of an external flash device.
Step 2	**format** {**slot0:** | **usb1:** | **usb2:**} **Example:** `switch# format slot0:`	Formats an external flash device.

Working with Directories

This section describes how to work with directories on the Cisco NX-OS device.

Identifying the Current Directory

You can display the directory name of your current directory.

Procedure

	Command or Action	Purpose
Step 1	**pwd** **Example:** `switch# pwd`	Displays the name of your current directory.

Changing the Current Directory

You can change the current directory for file system operations. The initial default directory is bootflash:.

Procedure

	Command or Action	Purpose
Step 1	**pwd** **Example:** `switch# pwd`	(Optional) Displays the name of your current default directory.
Step 2	**cd** {*directory* | *filesystem*:[//*module*/][*directory*]} **Example:** `switch# cd slot0:`	Changes to a new current directory. The file system, module, and directory names are case sensitive.

Creating a Directory

You can create directories in the bootflash: and flash device file systems.

Procedure

	Command or Action	Purpose
Step 1	**pwd** **Example:** `switch# pwd`	(Optional) Displays the name of your current default directory.
Step 2	**cd** {*directory* | *filesystem*:[//*module*/][*directory*]} **Example:** `switch# cd slot0:`	(Optional) Changes to a new current directory. The file system, module, and directory names are case sensitive.
Step 3	**mkdir** [*filesystem*:[//*module*/]]*directory* **Example:** `switch# mkdir test`	Creates a new directory. The *filesystem* argument is case sensitive. The *directory* argument is alphanumeric, case sensitive, and has a maximum of 64 characters.

Displaying Directory Contents

You can display the contents of a directory.

Procedure

	Command or Action	Purpose
Step 1	**dir** [*directory* | *filesystem*:[//*module*/][*directory*]] **Example:** `switch# dir bootflash:test`	Displays the directory contents. The default is the current working directory. The file system and directory names are case sensitive.

Deleting a Directory

You can remove directories from the file systems on your device.

Before You Begin

Ensure that the directory is empty before you try to delete it.

Procedure

	Command or Action	Purpose
Step 1	**pwd** **Example:** `switch# pwd`	(Optional) Displays the name of your current default directory.
Step 2	**dir** [*filesystem* :[//*module*/][*directory*]] **Example:** `switch# dir bootflash:test`	(Optional) Displays the contents of the current directory. The file system, module, and directory names are case sensitive. If the directory is not empty, you must delete all the files before you can delete the directory.
Step 3	**rmdir** [*filesystem* :[//*module*/]]*directory* **Example:** `switch# rmdir test`	Deletes a directory. The file system and directory name are case sensitive.

Accessing Directories on the Standby Supervisor Module

You can access all file systems on the standby supervisor module (remote) from a session on the active supervisor module. This feature is useful when copying files to the active supervisor modules requires similar files to exist on the standby supervisor module. To access the file systems on the standby supervisor module from a session on the active supervisor module, you specify the standby supervisor module in the path to the file using either *filesystem*:**//sup-remote/** or *filesystem*:**//sup-standby/**.

Working with Files

This section describes how to work with files on the Cisco NX-OS device.

Moving Files

You can move a file from one directory to another directory.

Caution If a file with the same name already exists in the destination directory, that file is overwritten by the moved file.

You can use the **move** command to rename a file by moving the file within the same directory.

Procedure

	Command or Action	Purpose
Step 1	**pwd** **Example:** `switch# pwd`	(Optional) Displays the name of your current default directory.
Step 2	**dir** [*filesystem*:[//*module*/][*directory*]] **Example:** `switch# dir bootflash`	(Optional) Displays the contents of the current directory. The file system and directory name are case sensitive.
Step 3	**move** [*filesystem*:[//*module*/][*directory* /] | *directory*/]*source-filename* {{*filesystem*:[//*module*/][*directory* /] | *directory*/}[*target-filename*] | *target-filename*} **Example:** `switch# move test old_tests/test1`	Moves a file. The file system, module, and directory names are case sensitive. The *target-filename* argument is alphanumeric, case sensitive, and has a maximum of 64 characters. If the *target-filename* argument is not specified, the filename defaults to the *source-filename* argument value.

Copying Files

You can make copies of files, either within the same directory or on another directory.

Note Use the **dir** command to ensure that enough space is available in the target file system. If enough space is not available, use the **delete** command to remove unneeded files.

Procedure

	Command or Action	Purpose
Step 1	**pwd** **Example:** `switch# pwd`	(Optional) Displays the name of your current default directory.
Step 2	**dir** [*filesystem*:[//*module*/][*directory*]] **Example:** `switch# dir bootflash`	(Optional) Displays the contents of the current directory. The file system and directory name are case sensitive.
Step 3	**copy** [*filesystem*:[//*module*/][*directory*/] | *directory*/]*source-filename* | {*filesystem*:[//*module*/][*directory*/]] | *directory*/}[*target-filename*] **Example:** `switch# move test old_tests/test1`	Copies a file. The file system, module, and directory names are case sensitive. The *source-filename* argument is alphanumeric, case sensitive, and has a maximum of 64 characters. If the *target-filename* argument is not specified, the filename defaults to the *source-filename* argument value.

Deleting Files

You can delete a file from a directory.

Procedure

	Command or Action	Purpose
Step 1	**dir** [*filesystem*:[//*module*/][*directory*]] **Example:** `switch# dir bootflash`	(Optional) Displays the contents of the current directory. The file system and directory name are case sensitive.
Step 2	**delete** {*filesystem*:[//*module*/][*directory*/] | *directory*/}*filename* **Example:** `switch# move test old_tests/test1`	Deletes a file. The file system, module, and directory names are case sensitive. The *source-filename* argument is case sensitive.

	Command or Action	Purpose
		Caution If you specify a directory, the **delete** command deletes the entire directory and all its contents.

Displaying File Contents

You can display the contents of a file.

Procedure

	Command or Action	Purpose
Step 1	**show file** [*filesystem*:[//*module*/]][*directory*/]*filename* **Example:** `switch# show file bootflash:test-results`	Displays the file contents.

Displaying File Checksums

You can display checksums to check the file integrity.

Procedure

	Command or Action	Purpose
Step 1	**show file** [*filesystem*:[//*module*/]][*directory*/]*filename* {**cksum** | **md5sum**} **Example:** `switch# show file bootflash:trunks2.cfg cksum`	Displays the checksum or MD5 checksum of the file.

Compressing and Uncompressing Files

You can compress and uncompress files on your Cisco NX-OS device using Lempel-Ziv 1977 (LZ77) coding.

Procedure

	Command or Action	Purpose
Step 1	**dir** [*filesystem*:[//*module*/]*directory*] **Example:** `switch# dir bootflash:`	(Optional) Displays the contents of the current directory. The file system and directory name are case sensitive.
Step 2	**gzip** [*filesystem*:[//*module*/][*directory*/] | *directory*/]*filename* **Example:** `switch# gzip show_tech`	Compresses a file. After the file is compressed, it has a .gz suffix.
Step 3	**gunzip** [*filesystem*:[//*module*/][*directory*/] | *directory*/]*filename* **.gz** **Example:** `switch# gunzip show_tech.gz`	Uncompresses a file. The file to uncompress must have the .gz suffix. After the file is uncompressed, it does not have the .gz suffix.

Displaying the Last Lines in a File

You can display the last lines of a file.

Procedure

	Command or Action	Purpose
Step 1	**tail** [*filesystem*:[//*module*/]][*directory*/]*filename* [*lines*] **Example:** `switch# tail ospf-gr.conf`	Displays the last lines of a file. The default number of lines is 10. The range is from 0 to 80 lines.

Redirecting show Command Output to a File

You can redirect **show** command output to a file on bootflash:, slot0:, volatile:, or on a remote server. You can also specify the format for the command output.

Procedure

	Command or Action	Purpose
Step 1	**terminal redirection-mode** {**ascii** | **zipped**} **Example:** `switch# terminal redirection-mode zipped`	(Optional) Set the redirection mode for the **show** command output for the user session. The default mode is **ascii**.
Step 2	*show-command* **>** [*filesystem***:**[//*module*/][*directory*] | [directory /]]*filename* **Example:** `switch# show tech-support > bootflash:techinfo`	Redirects the output from a **show** command to a file.

Finding Files

You can find the files in the current working directory and its subdirectories that have names that begin with a specific character string.

Procedure

	Command or Action	Purpose
Step 1	**pwd** **Example:** `switch# pwd`	(Optional) Displays the name of your current default directory.
Step 2	**cd** {*filesystem***:**[//*module*/][*directory*] | *directory*} **Example:** `switch# cd bootflash:test_scripts`	(Optional) Changes the default directory.
Step 3	**find** *filename-prefix* **Example:** `switch# find bgp_script`	Finds all filenames in the default directory and in its subdirectories beginning with the filename prefix. The filename prefix is case sensitive.

Working with Archive Files

The Cisco NX-OS software supports archive files. You can create an archive file, append files to an existing archive file, extract files from an archive file, and list the files in an archive file.

Creating an Archive Files

You can create an archive file and add files to it. You can specify the following compression types:

- bzip2
- gzip
- Uncompressed

The default is gzip.

Procedure

	Command or Action	Purpose
Step 1	**tar create** {**bootflash:** | **volatile:**}*archive-filename* [**absolute**] [**bz2-compress**] [**gz-compress**] [**remove**] [**uncompressed**] [**verbose**] *filename-list*	Creates an archive file and adds files to it. The filename is alphanumeric, not case sensitive, and has a maximum length of 240 characters. The **absolute** keyword specifies that the leading backslash characters (\) should not be removed from the names of the files added to the archive file. By default, the leading backslash characters are removed. The **bz2-compress**, **gz-compress**, and **uncompressed** keywords determine the compression utility used when files are added, or later appended, to the archive and the decompression utility to use when extracting the files. If you do not specify an extension for the archive file, the defaults are as follows: • For **bz2-compress**, the extension is .tar.bz2. • For **gz-compress**, the extension is .tar.gz. • For **uncompressed**, the extension is .tar. The **remove** keyword specifies that the Cisco NX-OS software should delete the files from the filesystem after adding them to the archive. By default, the files are not deleted. The **verbose** keyword specifies that the Cisco NX-OS software should list the files as they are added to the archive. By default, the files are listed as they are added.

This example shows how to create a gzip compressed archive file:

```
switch# tar create bootflash:config-archive gz-compress bootflash:config-file
```

Appending Files to an Archive File

You can append files to an existing archive file on your Cisco NX-OS device.

Before You Begin

You have created an archive file on your Cisco NX-OS device.

Procedure

	Command or Action	Purpose
Step 1	**tar append** {**bootflash:** | **volatile:**}*archive-filename* [**absolute**] [**remove**] [**verbose**] *filename-list*	Adds files to an existing archive file. The archive filename is not case sensitive. The **absolute** keyword specifies that the leading backslash characters (\) should not be removed from the names of the files added to the archive file. By default, the leading backslash characters are removed. The **remove** keyword specifies that the Cisco NX-OS software should delete the files from the filesystem after adding them to the archive. By default, the files are not deleted. The **verbose** keyword specifies that the Cisco NX-OS software should list the files as they are added to the archive. By default, the files are listed as they are added.

This example shows how to append a file to an existing archive file:

```
switch# tar append bootflash:config-archive.tar.gz bootflash:new-config
```

Extracting Files from an Archive File

You can extract files to an existing archive file on your Cisco NX-OS device.

Before You Begin

You have created an archive file on your Cisco NX-OS device.

Procedure

	Command or Action	Purpose
Step 1	**tar extract** {**bootflash:** | **volatile:**}*archive-filename* [**keep-old**] [**screen**] [**to** {**bootflash:** | **volatile:**}[*/directory-name*]] [**verbose**]	Extracts files from an existing archive file. The archive filename is not case sensitive. The **keep-old** keyword indicates that the Cisco NX-OS software should not overwrite files with the same name as the files being extracted. The **screen** keyword specifies that the Cisco NX-OS software should display the contents of the extracted files to the terminal screen. The **to** keyword specifies the target filesystem. You can include a directory name. The directory name is alphanumeric, case sensitive, and has a maximum length of 240 characters.

	Command or Action	Purpose
		The **verbose** keyword specifies that the Cisco NX-OS software should display the names of the files as they are extracted.

This example shows how to extract files from an existing archive file:

```
switch# tar extract bootflash:config-archive.tar.gz
```

Displaying the Filenames in an Archive File

You can display the names of the files in an archive files using the **tar list** command.

tar list {**bootflash:** | **volatile:**}*archive-filename*

The archive filename is not case sensitive.

```
switch# tar list bootflash:config-archive.tar.gz
config-file
new-config
```

Examples of Using the File System

This section includes example of using the file system on the Cisco NX-OS device.

Accessing Directories on Standby Supervisor Modules

This example shows how to list the files on the standby supervisor module:

```
switch# dir bootflash://sup-remote
   12198912     Aug 27 16:29:18 2003  m9500-sf1ek9-kickstart-mzg.1.3.0.39a.bin
    1864931     Apr 29 12:41:59 2003  dplug2
      12288     Apr 18 20:23:11 2003  lost+found/
   12097024     Nov 21 16:34:18 2003  m9500-sf1ek9-kickstart-mz.1.3.1.1.bin
   41574014     Nov 21 16:34:47 2003  m9500-sf1ek9-mz.1.3.1.1.bin

Usage for bootflash://sup-remote
   67747169 bytes used
  116812447 bytes free
  184559616 bytes total
```

This example shows how to delete a file on the standby supervisor module:

```
switch# delete bootflash://sup-remote/aOldConfig.txt
```

Moving Files

This example shows how to move a file on an external flash device:

```
switch# move slot0:samplefile slot0:mystorage/samplefile
```

This example shows how to move a file in the default file system:

```
switch# move samplefile mystorage/samplefile
```

Copying Files

This example shows how to copy the file called samplefile from the root directory of the slot0: file system to the mystorage directory:

```
switch# copy slot0:samplefile slot0:mystorage/samplefile
```

This example shows how to copy a file from the current directory level:

```
switch# copy samplefile mystorage/samplefile
```

This example shows how to copy a file from the active supervisor module bootflash to the standby supervisor module bootflash:

```
switch# copy bootflash:system_image bootflash://sup-2/system_image
```

This example shows how to overwrite the contents of an existing configuration in NVRAM:

```
switch# copy nvram:snapshot-config nvram:startup-config

Warning: this command is going to overwrite your current startup-config:
Do you wish to continue? {y/n} [y] y
```

You can also use the **copy** command to upload and download files from the slot0: or bootflash: file system to or from a FTP, TFTP, SFTP, or SCP server.

Deleting a Directory

You can remove directories from the file systems on your device.

Before You Begin

Ensure that the directory is empty before you try to delete it.

Procedure

	Command or Action	Purpose
Step 1	**pwd** **Example:** `switch# pwd`	(Optional) Displays the name of your current default directory.

	Command or Action	Purpose
Step 2	**dir** [*filesystem* **:**[*//module/*][*directory*]] **Example:** `switch# dir bootflash:test`	(Optional) Displays the contents of the current directory. The file system, module, and directory names are case sensitive. If the directory is not empty, you must delete all the files before you can delete the directory.
Step 3	**rmdir** [*filesystem* **:**[*//module/*]]*directory* **Example:** `switch# rmdir test`	Deletes a directory. The file system and directory name are case sensitive.

Displaying File Contents

This example displays the contents of a file on an external flash device:

```
switch# show file slot0:test
configure terminal
interface ethernet 1/1
no shutdown
end
show interface ethernet 1/1
```

This example displays the contents of a file residing in the current directory:

```
switch# show file myfile
```

Displaying File Checksums

This example shows how to display the checksum of a file:

```
switch# show file bootflash:trunks2.cfg cksum
583547619
```

This example shows how to display the MD5 checksum of a file:

```
switch# show file bootflash:trunks2.cfg md5sum
3b94707198aabefcf46459de10c9281c
```

Compressing and Uncompressing Files

This example shows how to compress a file:

```
switch# dir
    1525859     Jul 04 00:51:03 2003 Samplefile
...
switch# gzip volatile:Samplefile
switch# dir
```

```
     266069     Jul 04 00:51:03 2003 Samplefile.gz
...
```

This example shows how to uncompress a compressed file:

```
switch# dir
     266069     Jul 04 00:51:03 2003 Samplefile.gz
...
switch# gunzip samplefile
switch# dir
    1525859     Jul 04 00:51:03 2003 Samplefile
...
```

Redirecting show Command Output

This example shows how to direct the output to a file on the bootflash: file system:

```
switch# show interface > bootflash:switch1-intf.cfg
```

This example shows how to direct the output to a file on external flash memory:

```
switch# show interface > slot0:switch-intf.cfg
```

This example shows how to direct the output to a file on a TFTP server:

```
switch# show interface > tftp://10.10.1.1/home/configs/switch-intf.cfg
Preparing to copy...done
```

This example directs the output of the **show tech-support** command to a file:

```
switch# show tech-support > Samplefile
Building Configuration ...
switch# dir
    1525859     Jul 04 00:51:03 2003 Samplefile
Usage for volatile://
    1527808 bytes used
   19443712 bytes free
   20971520 bytes total
```

Finding Files

This example shows how to find a file in the current default directory:

```
switch# find smm_shm.cfg
/usr/bin/find: ./lost+found: Permission denied
./smm_shm.cfg
./newer-fs/isan/etc/routing-sw/smm_shm.cfg
./newer-fs/isan/etc/smm_shm.cfg
```

Default Settings for File System Parameters

This table lists the default settings for the file system parameters.

Table 20: Default File System Settings

Parameters	Default
Default filesystem	bootflash:

Additional References for File Systems

This section includes additional information related to the file systems.

Related Documents for File Systems

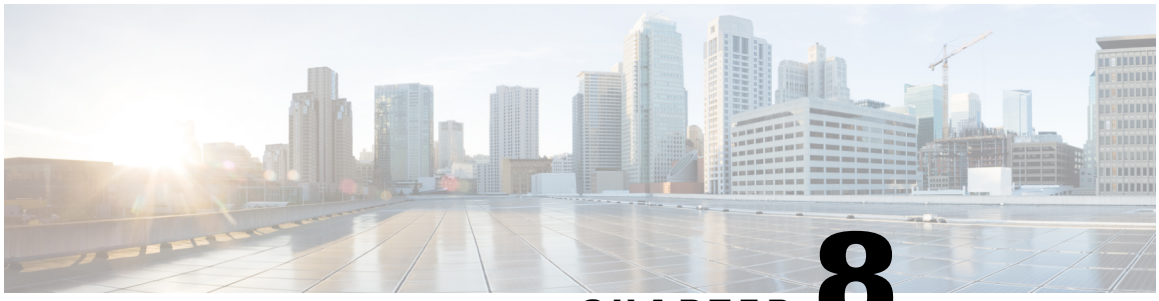
Related Topic	Document Title
Licensing	*Cisco NX-OS Licensing Guide*
Command reference	*Cisco Nexus 7000 Series NX-OS Fundamentals Command Reference*

Feature History for File Systems

This table lists the release history for this feature.

Table 21: Feature History for File Systems, Directories, and Files

Feature Name	Releases	Feature Information
show command output redirection	4.2(1)	You can change the format of the **show** command output when you redirect it to a file. The format can be ASCII or zipped.



CHAPTER 8

Working with Configuration Files

This chapter contains the following sections:

Information About Configuration Files

Configuration files contain the Cisco NX-OS software commands used to configure the features on a Cisco NX-OS device. Commands are parsed (translated and executed) by the Cisco NX-OS software when the system is booted (from the startup-config file) or when you enter commands at the CLI in a configuration mode.

To change the startup configuration file, you can either save the running-configuration file to the startup configuration using the **copy running-config startup-config** command or copy a configuration file from a file server to the startup configuration.

Types of Configuration Files

The Cisco NX-OS software has two types of configuration files, running configuration and startup configuration. The device uses the startup configuration (startup-config) during device startup to configure the software features. The running configuration (running-config) contains the current changes that you make to the startup-configuration file. The two configuration files can be different. You may want to change the device configuration for a short time period rather than permanently. In this case, you would change the running configuration by using commands in global configuration mode but not save the changes to the startup configuration.

To change the running configuration, use the **configure terminal** command to enter global configuration mode. As you use the Cisco NX-OS configuration modes, commands generally are executed immediately and are saved to the running configuration file either immediately after you enter them or when you exit a configuration mode.

To change the startup-configuration file, you can either save the running configuration file to the startup configuration or download a configuration file from a file server to the startup configuration.

Related Topics

Saving the Running Configuration to the Startup Configuration, on page 106

Downloading the Startup Configuration From a Remote Server, on page 109

Virtualization Support for Configuration Files

Except for removing the configuration for a missing module, the configuration file operations are local to the virtual device context (VDC). You can remove the missing module configuration only from the default VDC. For more information on VDCs, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide*.

Licensing Requirements for Configuration Files

The following table shows the licensing requirements for this feature:

Product	License Requirement
Cisco NX-OS	Configuration files require no license. Any feature not included in a license package is bundled with the Cisco NX-OS system images and is provided at no extra charge to you. For a complete explanation of the Cisco NX-OS licensing scheme, see the *Cisco NX-OS Licensing Guide*.

Managing Configuration Files

This section describes how to manage configuration files.

Saving the Running Configuration to the Startup Configuration

You can save the running configuration to the startup configuration to save your changes for the next time you that reload the device.

For information on saving the running configuration for all VDCs on the physical device, see the *Cisco Nexus 7000 Series NX-OS Virtual Device Context Configuration Guide*.

Procedure

	Command or Action	Purpose
Step 1	**show running-config** **Example:** `switch# show running-config`	(Optional) Displays the running configuration.
Step 2	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Copying a Configuration File to a Remote Server

You can copy a configuration file stored in the internal memory to a remote server as a backup or to use for configuring other Cisco NX-OS devices.

Procedure

	Command or Action	Purpose
Step 1	**copy running-config** *scheme*:*//server/*[*url /*]*filename* **Example:** `switch# copy running-config tftp://10.10.1.1/sw1-run-config.bak`	Copies the running-configuration file to a remote server. For the *scheme* argument, you can enter **tftp:**, **ftp:**, **scp:**, or **sftp:**. The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server. The *server*, *url*, and *filename* arguments are case sensitive.
Step 2	**copy startup-config** *scheme*:*//server/*[*url /*]*filename* **Example:** `switch# copy startup-config tftp://10.10.1.1/sw1-start-config.bak`	Copies the startup-configuration file to a remote server. For the *scheme* argument, you can enter **tftp:**, **ftp:**, **scp:**, or **sftp:**. The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server. The *server*, *url*, and *filename* arguments are case sensitive.

This example shows how to copy the configuration file to a remote server:

```
switch# copy running-config
tftp://10.10.1.1/sw1-run-config.bak
switch# copy startup-config
tftp://10.10.1.1/sw1-start-config.bak
```

Downloading the Running Configuration From a Remote Server

You can configure your Cisco NX-OS device by using configuration files that you created on another Cisco NX-OS device and uploaded to a remote server. You then download the file from the remote server to your device using TFTP, FTP, Secure Copy (SCP), or Secure Shell FTP (SFTP) to the running configuration.

Before You Begin

Ensure that the configuration file that you want to download is in the correct directory on the remote server.

Ensure that the permissions on the file are set correctly. Permissions on the file should be set to world-read.

Ensure that your Cisco NX-OS device has a route to the remote server. The Cisco NX-OS device and the remote server must be in the same subnetwork if you do not have a router or a default gateway to route traffic between subnets.

Check connectivity to the remote server using the **ping** or **ping6** command.

Procedure

	Command or Action	Purpose
Step 1	**copy** *scheme*:*//server/*[*url/*]*filename* **running-config** **Example:** `switch# copy tftp://10.10.1.1/my-config running-config`	Downloads the running-configuration file from a remote server. For the *scheme* argument, you can enter **tftp:**, **ftp:**, **scp:**, or **sftp:**. The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server. The *server*, *url*, and *filename* arguments are case sensitive.
Step 2	**show running-config** **Example:** `switch# show running-config`	(Optional) Displays the running configuration.
Step 3	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.
Step 4	**show startup-config** **Example:** `switch# show startup-config`	(Optional) Displays the startup configuration.

Related Topics

Copying Files, on page 101

Downloading the Startup Configuration From a Remote Server

You can configure your Cisco NX-OS device by using configuration files that you created on another Cisco NX-OS device and uploaded to a remote server. You then download the file from the remote server to your device using TFTP, FTP, Secure Copy (SCP), or Secure Shell FTP (SFTP) to the startup configuration.

Caution This procedure disrupts all traffic on the Cisco NX-OS device.

Before You Begin

Log in to a session on the console port.

Ensure that the configuration file that you want to download is in the correct directory on the remote server.

Ensure that the permissions on the file are set correctly. Permissions on the file should be set to world-read.

Ensure that your Cisco NX-OS device has a route to the remote server. The Cisco NX-OS device and the remote server must be in the same subnetwork if you do not have a router or a default gateway to route traffic between subnets.

Check connectivity to the remote server using the **ping** or **ping6** command.

Procedure

	Command or Action	Purpose
Step 1	**write erase** **Example:** `switch# write erase`	Erases the startup configuration file.

Step 2	**reload**
	
	Example:
	`switch# reload`
	`This command will reboot the system.`
	`(y/n)?  [n] y`
	`...`
	`Enter the password for "admin":`
	`<password>`
	`Confirm the password for "admin":`
	`<password>`
	`...`
	`Would you like to enter the basic configuration`
	`dialog (yes/no): n`
	`switch#`
		Reloads the Cisco NX-OS device.
		Note Do not use the setup utility to configure the device.

Step 3	**copy** *scheme***://***server*/[*url* /]*filename* **running-config** **Example:** `switch# copy tftp://10.10.1.1/my-config running-config`	Downloads the running configuration file from a remote server. For the *scheme* argument, you can enter **tftp:**, **ftp:**, **scp:**, or **sftp:**. The *server* argument is the address or name of the remote server, and the *url* argument is the path to the source file on the remote server.

	Command or Action	Purpose
		The *server*, *url*, and *filename* arguments are case sensitive.
Step 4	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Saves the running configuration file to the startup configuration file.
Step 5	**show startup-config** **Example:** `switch# show startup-config`	(Optional) Displays the running configuration.

Related Topics

Copying Files, on page 101

Copying Configuration Files to an External Flash Memory Device

You can copy configuration files to an external flash memory device as a backup for later use.

Before You Begin

Insert the external Flash memory device into the active supervisor module.

Procedure

	Command or Action	Purpose
Step 1	**dir** {**slot0:** | **usb1:** | **usb2:**}[*directory/*]	(Optional) Displays the files on the external flash memory device.
Step 2	**copy running-config** {**slot0:** | **usb1:** | **usb2:**}[*directory/*]*filename* **Example:** `switch# copy running-config slot0:dsn-running-config.cfg`	Copies the running configuration to an external flash memory device. The *filename* argument is case sensitive.
Step 3	**copy startup-config** {**slot0:** | **usb1:** | **usb2:**}[*directory/*]*filename* **Example:** `switch# copy startup-config slot0:dsn-startup-config.cfg`	Copies the startup configuration to an external flash memory device. The *filename* argument is case sensitive.

Related Topics

Copying Files, on page 101

Copying the Running Configuration From an External Flash Memory Device

You can configure your Cisco NX-OS device by copying configuration files created on another Cisco NX-OS device and saved to an external flash memory device.

Before You Begin

Insert the external flash memory device into the active supervisor module.

Procedure

	Command or Action	Purpose
Step 1	**dir** {**slot0:** | **usb1:** | **usb2:**}[*directory/*] **Example:** `switch# dir slot0:`	(Optional) Displays the files on the external flash memory device.
Step 2	**copy** {**slot0:** | **usb1:** | **usb2:**}[*directory/*]*filename* **running-config** **Example:** `switch# copy slot0:dsn-config.cfg running-config`	Copies the running configuration from an external flash memory device. The *filename* argument is case sensitive.
Step 3	**show running-config** **Example:** `switch# show running-config`	(Optional) Displays the running configuration.
Step 4	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.
Step 5	**show startup-config** **Example:** `switch# show startup-config`	(Optional) Displays the startup configuration.

Related Topics

Copying Files, on page 101

Copying the Startup Configuration From an External Flash Memory Device

You can recover the startup configuration on your Cisco NX-OS device by downloading a new startup configuration file saved on an external flash memory device.

Before You Begin

Insert the external flash memory device into the active supervisor module.

Procedure

	Command or Action	Purpose
Step 1	**dir** {**slot0:** | **usb1:** | **usb2:**}[*directory*/]	(Optional) Displays the files on the external flash memory device.
Step 2	**copy** {**slot0:** | **usb1:** | **usb2:**}[*directory*/]*filename* **startup-config** **Example:** `switch# copy slot0:dsn-config.cfg startup-config`	Copies the startup configuration from an external flash memory device. The *filename* argument is case sensitive.
Step 3	**show startup-config** **Example:** `switch# show startup-config`	(Optional) Displays the startup configuration.

Related Topics

Copying Files, on page 101

Copying Configuration Files to an Internal File System

You can copy configuration files to the internal memory as a backup for later use.

Procedure

	Command or Action	Purpose
Step 1	**copy running-config** [*filesystem***:**][*directory*/] | [*directory*/]*filename* **Example:** `switch# copy running-config bootflash:sw1-run-config.bak`	Copies the running-configuration file to internal memory. The *filesystem*, *directory*, and *filename* arguments are case sensitive.
Step 2	**copy startup-config** [*filesystem***:**][*directory*/] | [*directory*/]*filename*	Copies the startup-configuration file to internal memory.

	Command or Action	Purpose
	Example: `switch# copy startup-config bootflash:sw1-start-config.bak`	The *filesystem*, *directory*, and *filename* arguments are case sensitive.

Related Topics

Copying Files, on page 94

Rolling Back to a Previous Configuration

Problems, such as memory corruption, can occur that make it necessary for you to recover your configuration from a backed up version.

Note Each time that you enter a **copy running-config startup-config** command, a binary file is created and the ASCII file is updated. A valid binary configuration file reduces the overall boot time significantly. A binary file cannot be uploaded, but its contents can be used to overwrite the existing startup configuration. The **write erase** command clears the binary file.

Procedure

	Command or Action	Purpose
Step 1	**write erase** **Example:** `switch# write erase`	Clears the current configuration of the switch.
Step 2	**reload** **Example:** `switch# reload`	Restarts the device. You will be prompted to provide a kickstart and system image file for the device to boot and run.
Step 3	**copy** *configuration_file* **running-configuration** **Example:** `switch# copy bootflash:start-config.bak running-configuration`	Copies a previously saved configuration file to the running configuration. **Note** The *configuration_file* filename argument is case-sensitive.
Step 4	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the start-up configuration.

Removing the Configuration for a Missing Module

When you remove an I/O module from the chassis, you can also remove the configuration for that module from the running configuration. You can only remove the configuration for a missing module from the default VDC.

Note You can only remove the configuration for an empty slot in the chassis.

Before You Begin

Ensure that you are in the default VDC.

Remove the I/O module from the chassis.

Procedure

	Command or Action	Purpose
Step 1	**show hardware** **Example:** `switch# show hardware`	(Optional) Displays the installed hardware for the device.
Step 2	**purge module** *slot* **running-config** **Example:** `switch# purge module 3 running-config`	Removes the configuration for a missing module from the running configuration.
Step 3	**copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Erasing a Configuration

You can erase the configuration on your device to return to the factory defaults.

You can erase the following configuration files saved in the persistent memory on the device:

- Startup
- Boot
- Debug

Note The **write erase** command erases the entire startup configuration , except for the following:

- Boot variable definitions
- The IPv4 configuration on the mgmt0 interface, including the following:
  - Address
  - Subnet mask
  - Route address in the management VRF

To remove the boot variable definitions and the IPv4 configuration on the mgmt0 interface, use the **write erase boot** command.

Procedure

	Command or Action	Purpose

Step 1	**write erase** [**boot** | **debug**]
	
	Example:
	`switch# write erase`
	`Warning: This command will erase the startup-configuration.`
	`Do you wish to proceed anyway? (y/n) [n] y`
		Erases configurations in persistent memory. The default action erases the startup configuration.
		The **boot** option erases the boot variable definitions and the IPv4 configuration on the mgmt0 interface.
		The **debug** option erases the debugging configuration.
		Note The running configuration file is not affected by this command.

Clearing Inactive Configurations

You can clear inactive QoS and/or ACL configurations.

Procedure

	Command or Action	Purpose
Step 1	**show running-config** *type* **inactive-if-config** **Example:** `# show running-config ipqos inactive-if-config`	(Optional) Displays any inactive ACL or QoS configurations. The values for the *type* argument are **aclmgr** and **ipqos**. • **aclmgr**: Display any inactive configurations for aclmgr. • **ipqos**: Display any inactive configurations for qosmgr.

	Command or Action	Purpose

Step 2	**clear inactive-config** *policy*
	
	Example:
	`# clear inactive-config qos`
	`clear qos inactive config`
	`Inactive if config for QoS manager is saved at/bootflash/qos_inactive_if_config.cfg for vdc default & for other than default vdc: /bootflash/vdc_x/qos_inactive_if_config.cfg (where x is vdc number)`
	`you can see the log file @ show inactive-if-config log`
		Clears inactive configurations.
		
		The values for the *policy* argument are **qos** and **acl**.
		
		The following describes the values:
		• **qos**: Clear inactive QoS configurations.
		• **acl**: Clear inactive ACL configurations.
		• **acl qos**: Clear inactive ACL configurations and inactive QoS configurations.

Step 3	**show inactive-if-config log** **Example:** `# show inactive-if-config log`	(Optional) Displays the commands that were used to clear the inactive configurations.

Verifying the Device Configuration

To verify the configuration after bootstrapping the device using POAP, use one of the following commands:

Command	Purpose
show running-config	Displays the running configuration.
show startup-config	Displays the startup configuration.

For detailed information about the fields in the output from these commands, see the Cisco Nexus Command Reference for your device.

Examples of Working with Configuration Files

This section includes examples of working with configuration files.

Copying Configuration Files

This example shows how to overwrite the contents of an existing configuration in NVRAM:

```
switch# copy nvram:snapshot-config nvram:startup-config
Warning: this command is going to overwrite your current startup-config.
Do you wish to continue? {y/n} [y] y
```

This example shows how to copy a running configuration to the bootflash: file system:

```
switch# copy system:running-config bootflash:my-config
```

Backing Up Configuration Files

This example shows how to create a snapshot of the startup configuration in a predefined location on the device (binary file):

```
switch# copy startup-config nvram:snapshot-config
```

This example shows how to back up the startup configuration to the bootflash: file system (ASCII file):

```
switch# copy startup-config bootflash:my-config
```

This example shows how to back up the startup configuration to the TFTP server (ASCII file):

```
switch# copy startup-config tftp://172.16.10.100/my-config
```

This example shows how to back up the running configuration to the bootflash: file system (ASCII file):

```
switch# copy running-config bootflash:my-config
```

Rolling Back to a Previous Configuration

To roll back your configuration to a snapshot copy of a previously saved configuration, you need to perform the following steps:

1. Clear the current running image with the **write erase** command.
2. Restart the device with the **reload** command.
3. Copy the previously saved configuration file to the running configuration with the **copy** *configuration_file* **running-configuration** command.
4. Copy the running configuration to the start-up configuration with the **copy running-config startup-config** command.

Additional References for Configuration Files

This section includes additional information related to managing configuration files.

Related Documents for Configuration Files

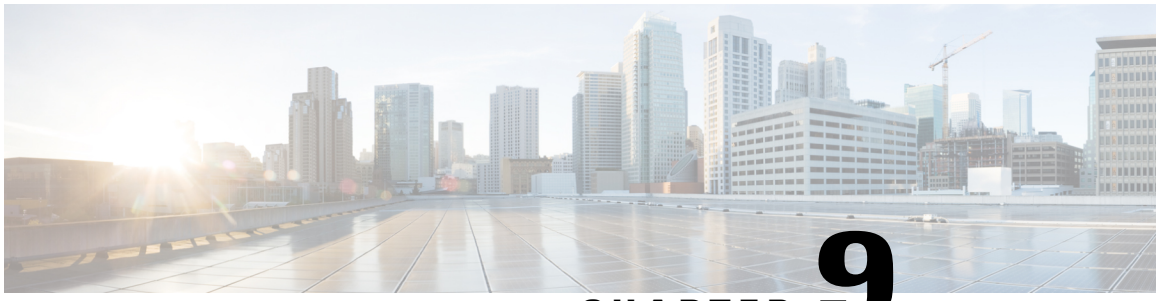
Related Topic	Document Title
Licensing	*Cisco NX-OS Licensing Guide*
Command reference	*Cisco Nexus 7000 Series NX-OS Fundamentals Command Reference*

Feature History for Configuration Files

This table lists the release history for this feature.

Table 22: Feature History for Configuration Files

Feature Name	Releases	Feature Information	
Configuration files	4.2(1)	This feature was introduced.	



CHAPTER 9

Scripting with Tcl

This chapter describes how to run tcl interactively and in scripts on a Cisco NX-OS device and includes the following sections:

Information about Tcl

Tcl (Tool Command Language) is a scripting language created by John Ousterhout at the University of California, Berkeley. Tcl 8.5 was added to Cisco NX-OS Release 5.1(1) to provide scripting abilities. With tcl, you gain more flexibility in your use of the CLI commands on the device. You can use tcl to extract certain values in the output of a **show** command, perform switch configurations, run Cisco NX-OS commands in a loop, or define EEM policies in a script.

This section describes how to run tcl scripts or run tcl interactively on Cisco NX-OS devices.

Guidelines and Limitations

Tcl has the following configuration guidelines and limitations:

Tclsh Command Help

Command help is not available for tcl commands. You can still access the help functions of Cisco NX-OS commands from within an interactive tcl shell.

This example shows the lack of tcl command help in an interactive tcl shell:

```
switch# tclsh
switch-tcl# set x 1
switch-tcl# puts ?
             ^
% Invalid command at '^' marker.
switch-tcl# configure ?
  <CR>
  session   Configure the system in a session
  terminal  Configure the system from terminal input

switch-tcl#
```

Note In the above example, the Cisco NX-OS command help function is still available but the tcl **puts** command returns an error from the help function.

Tclsh Command History

You can use the arrow keys on your terminal to access commands you previously entered in the interactive tcl shell.

Note The **tclsh** command history is not saved when you exit the interactive tcl shell.

Tclsh Tab Completion

You can use tab completion for Cisco NX-OS commands when you are running an interactive tcl shell. Tab completion is not available for tcl commands.

Tclsh CLI Command

Although you can directly access Cisco NX-OS commands from within an interactive tcl shell, you can only execute Cisco NX-OS commands in a tcl script if they are prepended with the tcl **cli** command.

In an interactive tcl shell, the following commands are identical and will execute properly:

```
switch-tcl# cli show module 1 | incl Mod
switch-tcl# cli "show module 1 | incl Mod"
switch-tcl# show module 1 | incl Mod
```

In a tcl script, you must prepend Cisco NX-OS commands with the tcl **cli** command as shown in the following example:

```
set x 1
cli show module $x | incl Mod
cli "show module $x | incl Mod"
```

If you use the following commands in your script, the script will fail and the tcl shell will display an error:

```
show module $x | incl Mod
"show module $x | incl Mod"
```

Tclsh Command Separation

The semicolon (:) is the command separator in both Cisco NX-OS and tcl. To execute multiple Cisco NX-OS commands in a tcl command, you must enclose the Cisco NX-OS commands in quotes ("").

In an interactive tcl shell, the following commands are identical and will execute properly:

```
switch-tcl# cli "configure terminal ; interface loopback 10 ; description loop10"
switch-tcl# cli configure terminal ; cli interface loopback 10 ; cli description loop10
switch-tcl# cli configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

switch(config-tcl)# cli interface loopback 10
```

```
switch(config-if-tcl)# cli description loop10
switch(config-if-tcl)#
```

In an interactive tcl shell, you can also execute Cisco NX-OS commands directly without prepending the tcl **cli** command:

```
switch-tcl# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

switch(config-tcl)# interface loopback 10
switch(config-if-tcl)# description loop10
switch(config-if-tcl)#
```

Tcl Variables

You can use tcl variables as arguments to the Cisco NX-OS commands. You can also pass arguments into tcl scripts. Tcl variables are not persistent.

This example shows how to use a tcl variable as an argument to a Cisco NX-OS command:

```
switch# tclsh
switch-tcl# set x loop10
switch-tcl# cli "configure terminal ; interface loopback 10 ; description $x"
switch(config-if-tcl)#
```

Tclquit

The **tclquit** command exits the tcl shell regardless of which Cisco NX-OS command mode is currently active. You can also press **Ctrl-C** to exit the tcl shell. The **exit** and **end** commands change Cisco NX-OS command modes. The **exit** command will terminate the tcl shell only from the EXEC command mode.

Tclsh Security

The tcl shell is executed in a sandbox to prevent unauthorized access to certain parts of the Cisco NX-OS system. The system monitors CPU, memory, and file system resources being used by the tcl shell to detect events such as infinite loops, excessive memory utilization, and so on.

You configure the intial tcl environment with the **scripting tcl init** *init-file* command.

You can define the looping limits for the tcl environment with the **scripting tcl recursion-limit** *iterations* command. The default recursion limit is 1000 interations.

Running the tclsh Command

You can run tcl commands from either a script or on the command line using the **tclsh** command.

Note You cannot create a tcl script file at the CLI prompt. You can create the script file on a remote device and copy it to the bootflash: directory on the Cisco NX-OS device.

Procedure

	Command or Action	Purpose

Step 1	**tclsh** [**bootflash:***filename* [*argument ...*]]
	
	Example:
	`switch# tclsh ?`
	`  <CR>`
	`  bootflash:  The file to run`
		Starts a tcl shell.
		If you run the **tclsh** command with no arguments, the shell runs interactively, reading tcl commands from standard input and printing command results and error messages to the standard output. You exit from the interactive tcl shell by typing **tclquit** or **Ctrl-C**.
		If you run the **tclsh** command with arguments, the first argument is the name of a script file containing tcl commands and any additional arguments are made available to the script as variables.

This example shows an interactive tcl shell:

```
switch# tclsh
switch-tcl# set x 1
switch-tcl# cli show module $x | incl Mod
Mod  Ports  Module-Type                       Model              Status
1    32     1/10 Gbps Ethernet Module        N7K-F132XP-15      ok
Mod  Sw              Hw
Mod  MAC-Address(es)                         Serial-Num
Mod  Online Diag Status
Left ejector CLOSE, Right ejector CLOSE, Module HW does support ejector based shutdown.
switch-tcl# exit
switch#
```

This example shows how to run a tcl script:

```
switch# show file bootflash:showmodule.tcl
set x 1
while {$x < 19} {
cli show module $x | incl Mod
set x [expr {$x + 1}]
}

switch# tclsh bootflash:showmodule.tcl
Mod  Ports  Module-Type                       Model              Status
1    32     1/10 Gbps Ethernet Module        N7K-F132XP-15      ok
Mod  Sw              Hw
Mod  MAC-Address(es)                         Serial-Num
Mod  Online Diag Status
Left ejector CLOSE, Right ejector CLOSE, Module HW does support ejector based shutdown.
switch#
```

Navigating Cisco NX-OS Modes from the tclsh Command

You can change modes in Cisco NX-OS while you are running an interactive tcl shell.

Procedure

	Command or Action	Purpose

Step 1	**tclsh**
	
	Example:
	`switch# tclsh`
	`switch-tcl#`
		Starts an interactive tcl shell.

Step 2	**configure terminal**
	
	Example:
	`switch-tcl# configure terminal`
	`switch(config-tcl)#`
		Runs a Cisco NX-OS command in the tcl shell, changing modes.
		
		Note The tcl prompt changes to indicate the Cisco NX-OS command mode.

Step 3	**tclquit**
	
	Example:
	`switch-tcl# tclquit`
	`switch#`
		Terminates the tcl shell, returning to the starting mode.

This example shows how to change Cisco NX-OS modes from an interactive tcl shell:

```
switch# tclsh
switch-tcl# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config-tcl)# interface loopback 10
switch(config-if-tcl)# ?
  description  Enter description of maximum 80 characters
  inherit      Inherit a port-profile
  ip           Configure IP features
  ipv6         Configure IPv6 features
  logging      Configure logging for interface
  no           Negate a command or set its defaults
  rate-limit   Set packet per second rate limit
  shutdown     Enable/disable an interface
  this         Shows info about current object (mode's instance)
  vrf          Configure VRF parameters
  end          Go to exec mode
  exit         Exit from command interpreter
  pop          Pop mode from stack or restore from name
  push         Push current mode to stack or save it under name
  where        Shows the cli context you are in

switch(config-if-tcl)# description loop10
switch(config-if-tcl)# tclquit
Exiting Tcl
switch#
```

Tcl References

The following titles are provided for your reference:

- Mark Harrison (ed), *Tcl/Tk Tools*, O'Reilly Media, ISBN 1-56592-218-2, 1997

- Mark Harrison and Michael McLennan, *Effective Tcl/Tk Programming*, Addison-Wesley, Reading, MA, USA, ISBN 0-201-63474-0, 1998
- John K. Ousterhout, *Tcl and the Tk Toolkit*, Addison-Wesley, Reading, MA, USA, ISBN 0-201-63337-X, 1994.
- Brent B. Welch, *Practical Programming in Tcl and Tk*, Prentice Hall, Upper Saddle River, NJ, USA, ISBN 0-13-038560-3, 2003.
- J Adrian Zimmer, *Tcl/Tk for Programmers*, IEEE Computer Society, distributed by John Wiley and Sons, ISBN 0-8186-8515-8, 1998.

INDEX

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V