

Cisco Nexus 9000 Series NX-OS System Management Configuration Guide, Release 10.6(x)

First Published: 2025-08-13

Last Modified: 2026-04-27

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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS REFERENCED IN THIS DOCUMENTATION ARE SUBJECT TO CHANGE WITHOUT NOTICE. EXCEPT AS MAY OTHERWISE BE AGREED BY CISCO IN WRITING, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENTATION ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

The Cisco End User License Agreement and any supplemental license terms govern your use of any Cisco software, including this product documentation, and are located at: https://www.cisco.com/c/en/us/about/legal/cloud-and-software/software-terms.html. Cisco product warranty information is available at https://www.cisco.com/c/en/us/products/warranty-listing.html. US Federal Communications Commission Notices are found here https://www.cisco.com/c/en/us/products/us-fcc-notice.html.

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.

Any products and features described herein as in development or available at a future date remain in varying stages of development and will be offered on a when-and if-available basis. Any such product or feature roadmaps are subject to change at the sole discretion of Cisco and Cisco will have no liability for delay in the delivery or failure to deliver any products or feature roadmap items that may be set forth in this document.

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

The documentation set for this product strives to use bias-free language. For the purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

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: https://www.cisco.com/c/en/us/about/legal/trademarks.html. 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. (1721R)

© 2025–2026 Cisco Systems, Inc. All rights reserved.



CONTENTS

Trademarks ?

PREFACE

Preface xxi

Audience xxi

Document Conventions xxi

Related Documentation for Cisco Nexus 9000 Series Switches xxii

Documentation Feedback xxii

Communications, services, and additional information xxii

Cisco Bug Search Tool xxiii

Documentation feedback xxiii

CHAPTER 1

New and Changed Information 1

New and Changed Information 1

CHAPTER 2

Overview 7

Overview 7

CHAPTER 3

Two-stage Configuration Commit 13

Two-stage configuration commit process 13

Guidelines and limitations for two-stage configuration commit 14

Configuring in Two-Stage Configuration Commit Mode 15

Abort the two-stage configuration commit mode 20

Display commit IDs 21

Rollback capability 21

View current session configurations 22

CHAPTER 4 **Switch Profiles** 23

Overview 23

Guidelines and Limitations 25

Configure Switch Profile 27

Add or Modify Switch Profile Commands 29

Import Switch Profile 30

Import Configurations in vPC Topology 32

Isolate Peer Switch 33

Delete Switch Profile 33

Manual Correction for Mutex and Merge Failures 34

Migrate to Cisco NX-OS Release 7.0(3)I2(1) or Higher in a Fabric Extender Straight-Through Topology 34

Replace Cisco Nexus 9000 Series Switch 35

Synchronize Configurations After Cisco Nexus 9000 Series Switch Reboots 36

Synchronize Configurations When mgmt0 Interface Connectivity Is Lost 36

Revert Inadvertent Port Mode Change of Layer 2 to Layer 3 in Global Configuration Mode 37

Verify Switch Profile Configuration 37

Configuration Examples for Switch Profiles 38

CHAPTER 5 **Configuring Frequency Synchronization** 49

Frequency synchronization in networks 49

Hybrid SyncE-PTP with external PRC source 50

Timing sources 50

Timing inputs 50

Timing outputs 51

Timing source selection points 51

Licensing requirements for Synchronous Ethernet (SyncE) 52

Guidelines and limitations for Frequency Synchronization 52

Configure Frequency Synchronization 53

Enable Frequency Synchronization 53

Configure Frequency Synchronization on an interface 55

Commands for verifying frequency synchronization configuration 57

CHAPTER 6 **Precision Time Protocol** 61

Functionality 61
Default PTP Parameters 63
Clocks 64
Generalized Precision Time Protocol 65
Process 66
ITU-T Telecom Profiles 67
Profile Defaults 68
Telecom Profile G.8275.1 69
Telecom Profile G.8275.2 70
Guidelines and limitations for PTP 72
Configure PTP 80
Configure PTP Globally 80
Configure PTP GM 84
Configure PTP on an Interface 85
Configure PTP in Unicast Mode 87
Configure Unicast Mode 87
Configure Unicast Mode for IPv4 or IPv6 88
Assign Dynamic Role 89
Assign Master Role 90
Assign Slave Role 92
Configure Unicast Source Address 94
Configure PTP Telecom Profile 94
Configure Global PTP Telecom Profile 94
Configure PTP Telecom Profile 8275-1 on an Interface 97
Configure PTP Telecom Profile 8275-2 on an Interface 99
Verify PTP Telecom Profile Configuration 102
Configure PTP Notifications 106
Configure PTP Interface to Stay in Master State 108
Verify PTP Configuration 110
Configuration Examples 111
PTP Unicast-Negotiation 113
Enable PTP Unicast-Negotiation 114

Enhanced Multicast Scale 115
Timestamp Tagging 115
Configure Timestamp Tagging 116
Configure TTAG Marker Packets and Time Interval 117
Related Documents 119

CHAPTER 7 Configuring GPS 121
Global Positioning System 121
Guidelines and limitations for GPS 121
Configure GPS settings 122
Commands for verifying GPS configuration 123

CHAPTER 8 Network Time Protocol 125
Overview 125
Guidelines and Limitations 127
Enable or Disable NTP 130
Configure Device as an Authoritative NTP Server 133
Configure Server and Peer 133
Configure Authentication 135
Configure Access Restrictions 137
Configure Source IP Address 139
Configure Source Interface 139
Configure Logging 140
Additional References 141

CHAPTER 9 Configuring GNSS 143
Global Navigation Satellite System 143
Guidelines and limitations for GNSS 144
Enable the GNSS receiver 144
Commands for verifying GNSS configuration 146

CHAPTER 10 Cisco Discovery Protocol 149
Overview 149
Guidelines and Limitations 151

Enable or Disable Globally 151
Enable or Disable on an Interface 153
Configure Optional CDP Parameters 154

CHAPTER 11 System Message Logging 155
Overview 155
Guidelines and Limitations 157
Configure System Message Logging to Terminal Sessions 159
Configure Origin ID for Syslog Messages 161
Log System Messages to a File 162
Configure Module and Facility Messages Logging 164
Configure Logging Syslogs Compliant to RFC 5424 166
Configure Syslog Servers 167
Configure Secure Syslog Server 169
TCP Keepalive for Syslog Logging 170
Guidelines and Limitations for TCP Keepalive 171
Configure TCP Keepalive for Syslog Logging 171
Configure CA Certificate 172
Enroll CA Certificate 173
Configure Syslog Server on a UNIX or Linux System 174
Display and Clear Log Files 175
Configuration for System Message Logging 177
Related Documents 178

CHAPTER 12 Configure Audit Log Reporting 179
AuditD 179
Guidelines and Limitations 179
Configure AuditD 180
Monitor Rules 181
Verifying AuditD Configuration 182

CHAPTER 13 Smart Call Home 183
Overview 183
Guidelines and Limitations 190

Configuration 191
Configure Contact Information 194
Create Destination Profile 195
Modify Destination Profile 197
Associate Alert Group with Destination Profile 199
Add Show Commands to Alert Group 200
Configure Email Server 201
Configure VRFs to Send Messages Using HTTP 202
Configure HTTP Proxy Server 203
Configure Periodic Inventory Notifications 205
Disable Duplicate Message Throttling 206
Enable or Disable Smart Call Home 207
Configure SMTP-AUTH for Call Home Mail Transfer 208
Configure Source Interface 210
Test Smart Call Home Configuration 210
Additional Reference Information 211

CHAPTER 14 **Session Manager** 225
Session Manager 225
High Availability 226
Prerequisites for Session Manager 226
Guidelines and limitations for Session Manager 226
Configure Session Manager 226
Create a session 226
Configure ACLs in a session 227
Verify a session 228
Commit a session 228
Save a session 229
Discard a Session 229
Commands for verifying Session Manager configuration 229
Configuration example for Session Manager 229
Additional references 230
Related documents 230

CHAPTER 15 **Scheduler** 231

Overview 231
Guidelines and Limitations 232
Enable or Disable the Scheduler 233
Define Scheduler Log File Size 234
Configure Remote User Authentication 235
Define Job 236
Delete Job 237
Define Timetable 237
Clear Scheduler Log File 239
Verify Scheduler Configuration 240
Configuration Examples 240

CHAPTER 16 **Simple Network Management Protocol** 243

SNMP Fundamentals 243
SNMPv3 247
Disable Security and SNMP User Synchronization 249
Guidelines and Limitations 250
Configure SNMP 253
Configure SNMP Users 253
Generate Hashed Password Offline 255
Enforce SNMP Message Encryption 255
Assign SNMPv3 Users to Multiple Roles 256
Create SNMP Communities 257
Filter SNMP Requests 257
Configure SNMP Notification Receivers 258
Configure Source Interface for SNMP Notifications 259
Configure Notification Target User 261
Configure SNMP Notification Receivers with VRFs 262
Configure SNMP to Send Traps Using an Inband Port 263
Enable SNMP Notifications 264
Disable Link Notifications on an Interface 272
Display SNMP ifIndex for an Interface 273

Enable One-Time Authentication for SNMP over TCP 273
Assign SNMP Device Contact and Location Information 274
Configure Context to Network Entity Mapping 275
Disable SNMP 276
Modify SNMP Server Counter Cache Update Timer 277
Modify AAA Synchronization Time 277
Configure SNMP Local Engine ID 278
Verify SNMP Configuration 279
Configuration Examples 280
SNMP Sample Outputs 282
Related Documents 283

CHAPTER 17 **Remote Monitoring (RMON)** 285
Overview 285
Guidelines and Limitations 287
Configuration 287
Configure Alarms 288
Configure Events 289
Additional References 290

CHAPTER 18 **Online Diagnostics** 291
Generic Online Diagnostics 291
Bootup diagnostics 292
Runtime or health monitoring diagnostics 293
On-demand diagnostics 298
High Availability 298
Virtualization support 298
Guidelines and limitations for online diagnostics 299
Default settings for online diagnostics 300
Configuring Online Diagnostics 300
Set the bootup diagnostic level 300
Activate a diagnostic test 301
Start or stop an on-demand diagnostic test 302
Simulate diagnostic results 303

Clear diagnostic results 304
Commands for verifying Online Diagnostics configuration 304
Configuration examples for Online Diagnostics 305

CHAPTER 19 Configuring the Embedded Event Manager 307
Embedded Event Manager 307
Policies 308
Event statements 309
Action statements 309
VSH script policies 310
Environment variables 311
EEM event correlation 311
High Availability 311
Virtualization support 311
Prerequisites for Embedded Event Manager 311
Guidelines and limitations for Embedded Event Manager 311
Default settings for EEM 314
Configure Embedded Event Manager 314
Defining an Environment Variable 314
Define a user policy using the command line interface 315
Configure event statements 316
Configure action statements 320
Define a policy using a VSH script 322
Register and activate a VSH script policy 323
Override a policy 323
Configure memory thresholds 324
Configure syslog as EEM publisher 326
Commands for verifying the Embedded Event Manager configuration 327
Configuration examples for Embedded Event Manager 327
Event Log Auto-Collection and backup 329
Extended Log File Retention 329
Enable Extended Log File Retention for all services 329
Disable Extended Log File Retention for all services 329
Enable Extended Log File Retention for a single service 330

Display Extended Log Files 331
Display global dictionary per log statistics 332
Disable Extended Log File Retention for a single service 332
Trigger-based event log auto-collection 333
Enable trigger-based log file auto-collection 334
Log-profile YAML files 334
Auto-Collection YAML files 335
Auto-collection bundle limits per component 342
Auto-Collection log files 342
Examples of supported log collection 346
Verify trigger-based log collection 346
Check trigger-based log file generation 347
Local log file storage 347
Generate a local copy of recent log files 347
External log file storage 349

CHAPTER 20 **Configuring MAC Move Policy** 353
MAC move policy 353
Guidelines and Limitations for MAC Move Policy 353
Configuring MAC Move Policy 354
Verifying the MAC Move Policy Configuration 355

CHAPTER 21 **Terminal Lock for VSH Sessions** 357
Overview 357

CHAPTER 22 **Onboard Failure Logging** 361
Overview 361
Guidelines and Limitations 362
Configuration 362
Related Documents 365

CHAPTER 23 **Switched Port Analyzer** 367
Ports 367
Sessions 369

SPAN Functionality 370
Guidelines and Limitations 370
Cisco Nexus 9200 Platform Switches 375
Cisco Nexus 9300 Platform Switches 376
Cisco Nexus 9336C-SE1 switch 378
Cisco Nexus 9364E-SG2 switches 379
N9300 Smart Switches 379
Cisco Nexus 9500 Platform Switches 380
Cisco Nexus 9800 Platform Switches 382
Configure SPAN Session 383
Configure UDF-Based SPAN 386
Configure SPAN Truncation 389
Configure SPAN for Multicast Tx Traffic Across Different LSE Slices 391
Configure SPAN to CPU 393
Shut Down or Resume SPAN Session 395
Verify SPAN Configuration 397
Configuration Examples 397
Related Documents 398

CHAPTER 24 **Encapsulated Remote Switched Port Analyzer** 399
Overview 399
Guidelines and Limitations 401
Configure an ERSPAN Source Session 408
Configure an ERSPAN Destination Session 412
Configure an ERSPAN ACL 414
Configure UDF-Based ERSPAN 417
Configure ERSPAN Truncation 420
Configuration Examples 422

CHAPTER 25 **Link Layer Discovery Protocol** 425
Overview 425
Guidelines and Limitations 427
Enable or Disable Globally 429
Enable or Disable on an Interface 431

Configure DCBXP Egress Queuing 432
Configure DCBXP Protocol Version 433
Enable or Disable Multi-Neighbor Support 434
Enable or Disable Support on Port-Channel Interfaces 437
Configure Optional Parameters 439

CHAPTER 26 NetFlow 441
NetFlow Fundamentals 442
Ingress NetFlow 445
Configure Ingress NetFlow 451
Configuration Example for Ingress NetFlow 451
Ingress NetFlow Verification Commands 451
Egress NetFlow 453
Configure Egress NetFlow 454
Configuration Example for Egress NetFlow 454
Egress NetFlow Verification Commands 455
Sampled Netflow 455
General Procedures to Enable NetFlow 457
Enable the NetFlow feature 457
Create a flow record 458
Specify match parameters 459
Specify collect parameters 460
Create a flow exporter 460
Create Flow Monitor 462
Apply Flow Monitor to an Interface 463
Additional Procedures for Ingress NetFlow 464
Configure Bridged NetFlow on VLAN 464
Configure Layer 2 NetFlow Keys 465
Configure Layer 3 NetFlow on Layer 2 Interfaces 467
Configure NetFlow Timeouts 468
Additional Procedures for Egress NetFlow 468
Drop Reports 469
Configure Drop Report 470
Verify Drop Report Configuration 471

Dropped flow notification 472
Configure Dropped Flow Notification 472
Sample configuration for dropped flow notification 474
Verification commands for dropped flow notification 474

CHAPTER 27 Mixed Mode 475
Mixed Mode 475
Guidelines and limitations for Mixed Mode 475
Mixed mode use cases 476
Switches already deployed with feature analytics 477
Switches already deployed with feature netflow 477
Switches that have neither features configured 478
Commands for verifying the Mixed Mode configuration 479
Display example for Mixed Mode 480

CHAPTER 28 Configuring Traffic Analytics 483
Traffic Analytics 483
Aging of traffic database entries 483
Troubleshooting rules 484
Faster export cadence for troubleshoot flow records 484
UDP port support and configuration 484
Traffic Analytics interface filter and VRF filter 485
Interface Traffic Analytics 485
ECN detection for Traffic Analytics 487
How ECN Detection Works 487
Disable global Traffic Analytics 488
Guidelines and limitations for Traffic Analytics 488
Configure Traffic Analytics 490
Example for Traffic Analytics interface filter and VRF filter 491
Example for Traffic Analytics 493

CHAPTER 29 Sampled Flow 495
Overview 495
Guidelines and Limitations 496

Enable sFlow Feature 500
Configure Sampling Rate 502
Configure Maximum Sampled Size 502
Configure Counter Poll Interval 503
Configure Maximum Datagram Size 504
Configure sFlow Collector Address 505
Configure Collector Port 506
Configure Agent Address 507
Configure Sampling Data Source 508
Configure Extended BGP (Gateway) 509
Related Documents 509

CHAPTER 30 **Test Access Points Aggregation** 511
About TAP Aggregation 511
Enable TAP Aggregation for Line Cards 514
Configure TAP Aggregation Policy 515
Attach TAP Aggregation Policy to Interface 516
Configure Selective Q-in-Q with Provider VLANs 517
About MPLS Stripping 520
Enable MPLS Stripping 521
Configure Incoming Port for VLAN Tag 522
Add and Delete MPLS Labels 523
Configure Destination MAC Address 524
Configure MPLS Label Aging 525
MPLS Stripping Configuration Commands and Examples 526
Related Documents 528

CHAPTER 31 **Packet truncation** 529
Configure packet truncation at the egress ports task 531
Configure packet truncation with CRC stomping disabled 531
Configure packet truncation to exclude control packets 532
Configure packet truncation to exclude control packets at system level 533
Configure packet truncation to exclude control packets on an interface 533
Configure system-wide packet truncation CRC check 534

Verify packet truncation 535

CHAPTER 32 **Configuring Deduplication** 537

Deduplication concept and models 537

Guidelines and limitations for deduplication 538

Configure deduplication 539

Monitor commands for deduplication 540

CHAPTER 33 **MPLS Access Lists** 543

Configure MPLS access lists 543

Verification of MPLS access lists configuration 544

Configuration example for MPLS access lists 544

CHAPTER 34 **Header Stripping for Nexus Data Broker** 545

Overview 545

Guidelines and Limitations 547

VXLAN and iVXLAN Header Stripping 548

Overview 548

Guidelines and Limitations 548

Configure Nexus Data Broker Termination 550

ERSPAN Header Stripping 552

Overview 552

Guidelines and Limitations 553

Configure ERSPAN Header Stripping 554

GRE Header Stripping 556

Overview 556

Guidelines and Limitations 556

Configuration 557

MPLS Header Stripping 559

Overview 559

Guidelines and Limitations 559

Configuration 561

CHAPTER 35 **Graceful Insertion and Removal** 565

Overview 565
Guidelines and Limitations 567
Configure Maintenance-Mode Profile 568
Configure Normal-Mode Profile 571
Create Snapshot 572
Add Show Commands to Snapshots 574
Trigger Graceful Removal 575
Trigger Graceful Insertion 578
Maintenance Mode Enhancements 579
Verify Configuration 580
Configuration Examples 581

CHAPTER 36 **Software Maintenance Upgrades** 583
Overview 583
Guidelines and Limitations 585
Prepare for Package Installation 586
Download SMU Package File from Cisco.com 588
Copy Package File to Local Storage Device or Network Server 588
Add and Activate Package 590
Commit Active Package Set 592
Deactivate and Remove Packages 593
No-Reload Options for SMU Installation 595
Advanced SMU Installation Methods 600
Downgrade Feature RPMs 603
Display Installation Log Information 605
Perform Software Maintenance Upgrade for Guest Shell Bash 605
Related Documents 606
History 607

CHAPTER 37 **Configuration Replace and Commit-timeout** 609
Overview 609
Guidelines and Limitations 618
Perform Configuration Replace 622

CHAPTER 38 **Rollback** 625

Overview 625

Prerequisites for Rollbacks 626

Guidelines and Limitations 627

Configuration 628

Create Checkpoint 628

Implement Rollback 629

Additional References 630

CHAPTER 39 **Candidate Configuration** 631

Overview 631

Guidelines and Limitations 633

Perform Integrity Check for Candidate Config 638

Configuration Candidate session 639

Guidelines and Limitations for Configuration Candidate Session 640

Configuring a Candidate Session 640

Verifying a Candidate Session 642

CHAPTER 40 **Secure Erase** 645

Secure erase 645

Guidelines and limitations for secure erase 646

Perform a secure erase 646

Secure erase examples 648

APPENDIX A **IETF RFCs supported by Cisco NX-OS System Management** 657

IETF RFCs supported by Cisco NX-OS System Management 657

APPENDIX B **Embedded Event Manager System Events and Configuration Examples** 659

System Policies 659

Events 661

Configuration Examples 662

Command Line Interface Events 662

Override (Disable) Major Thresholds 663

Override (Disable) Shutdown for Fan Tray Removal 667
Create Supplemental Policy 669
Power Over-Budget Policy 669
Select Modules to Shut Down 670
Online Insertion Removal Event 671
Generate User Syslog 671
Monitor Syslog Messages 671
SNMP Notification 672
Port Tracking 672
Register an EEM Policy with the EEM 673

APPENDIX C System Management Configuration Limits in Cisco NX-OS 677
System management configuration limits in Cisco NX-OS 677



Preface

This preface includes the following sections:

- Audience, on page xxi
- Document Conventions, on page xxi
- Related Documentation for Cisco Nexus 9000 Series Switches, on page xxii
- Documentation Feedback, on page xxii
- Communications, services, and additional information, on page xxii

Audience

This publication is for network administrators who install, configure, and maintain Cisco Nexus switches.

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 you supply the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments that are separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments that are 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.

Convention	Description
`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 includes 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 that 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.

Related Documentation for Cisco Nexus 9000 Series Switches

The entire Cisco Nexus 9000 Series switch documentation set is available at the following URL:

https://www.cisco.com/en/US/products/ps13386/tsd_products_support_series_home.html

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus9k-docfeedback@cisco.com. We appreciate your feedback.

Communications, services, and additional information

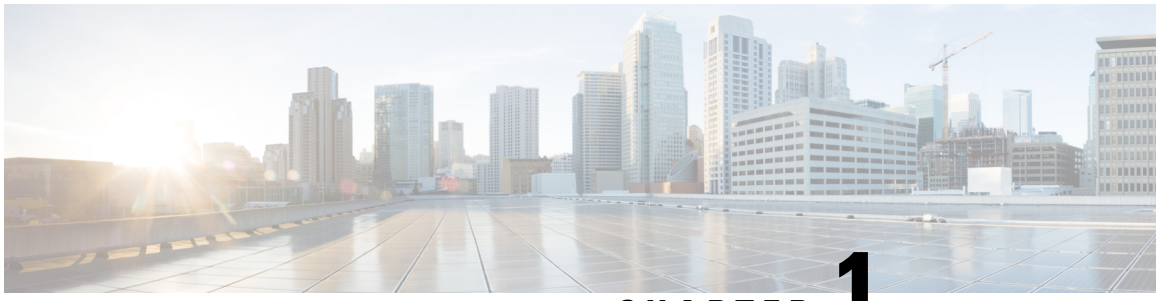
- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions, and services, visit Cisco DevNet.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a gateway to the Cisco bug-tracking system, which maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. The BST provides you with detailed defect information about your products and software.

Documentation feedback

To provide feedback about Cisco technical documentation, use the feedback form available in the right pane of every online document.



CHAPTER 1

New and Changed Information

- New and Changed Information, on page 1

New and Changed Information

This table summarizes the new and changed features for the Nexus 9000 Series NX-OS System Management Configuration Guide, Release 10.6(x) and where they are documented.

Table 1: New and Changed Features

Feature	Description	Changed in Release	Where Documented
FIPS MACSec Certification	The bootup diagnostic process is enhanced by introducing a mandatory security check for C9364D-GX2A, C9348D-GX2A and C9332D-GX2B hardware platforms.	10.6(3)F	Bootup Diagnostics
Configuration Candidate Session	Allows you to stage, review, and validate configuration changes before committing them to the running configuration on Cisco Nexus 9000 Series Switches.	10.6(3)F	Configuration Candidate Session
Exclude control packets from packet truncation	Provided option to exclude control packets from truncation at the system level or at the interface level	10.6(3)F	Guidelines and limitations for packet truncation, on page 530 Configure packet truncation to exclude control packets, on page 532

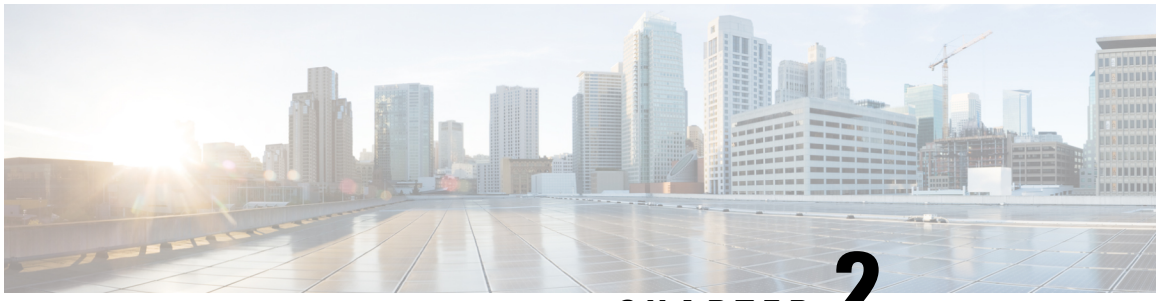
Feature	Description	Changed in Release	Where Documented
PTP support on N9300 Smart switches	PTP support is added on Cisco N9324C-SE1U and N9348Y2C6D-SE1U Smart switches.	10.6(3)F	Guidelines and limitations for PTP, on page 72
Syslogd console logging performance improvement	Improved syslogd responsiveness by handling console logging in the background so slow or unresponsive console output does not block syslog processing. Console messages may be dropped if internal buffering is full. Added console-specific drop/error statistics to **show logging dropcount** and **clear logging dropcount**.	10.6(3)F	Guidelines and Limitations
TCP Keepalive for Syslog Logging	Added support for configurable TCP keepalive idle timeout on syslog TCP connections to maintain reliable log delivery and prevent session drops due to idle network timeouts.	10.6(3)F	TCP Keepalive for Syslog Logging
GOLD support on N9164E-NS4-O switch	Added support for Generic On-Line Diagnostics (GOLD) on N9164E-NS4-O switch	10.6(2n)	Guidelines and limitations for online diagnostics, on page 299
SNMP support on N9164E-NS4-O switch	Added support for SNMP on N9164E-NS4-O switch	10.6(2n)	Guidelines and Limitations, on page 250
GIR support on N9300 Smart Switches	Added support for Graceful Insertion and Replacement (GIR) on N9324C-SE1U and N9348Y2C6D-SE1U Smart Switches	10.6(2)F	N9300 Smart Switches Guidelines, on page 568

Feature	Description	Changed in Release	Where Documented
Sampled Netflow support on N9364E-SG2-O and N9364E-SG2-Q switches	Added support for packet-level traffic sampling for IPv4 and IPv6 flows to collect representative network data at a configurable sampling rate.	10.6(2)F	Sampled Netflow, on page 455 Create Flow Monitor, on page 462
Full netflow version 9 support on 9300-SE1 and 9300-SE1U switches	Added support for ingress NetFlow on N9336C-SE1, N9396T12C-SE1, N9396Y12C-SE1, N9348Y12C-SE1, N9324C-SE1U, and N9348Y2C6D-SE1U switches.	10.6(2)F	Ingress NetFlow, on page 445 Egress NetFlow, on page 453
AuditD Implementation	Added support for AuditD	10.6(2)F	Guidelines and Limitations, on page 179 Configure AuditD, on page 180
SNMPv1, SNMPv2c, and SNMPv3 support on N9300 Smart Switches	Added support for SNMPv1, SNMPv2c, and SNMPv3 on N9324C-SE1U and N9348Y2C6D-SE1U Smart Switches	10.6(2)F	Cisco N9300 Smart Switches, on page 253
sFlow support on N9300 Smart Switches	Added support for sampled Flow on N9324C-SE1U and N9348Y2C6D-SE1U Smart Switches	10.6(2)F	N9300 Smart Switch Guidelines, on page 500
LLDP support on N9300 Smart Switches	Added support for LLDP on N9324C-SE1U and N9348Y2C6D-SE1U Smart Switches	10.6(2)F	N9300 Series Smart Switch Guidelines, on page 429
ERSPAN support on N9300 Smart Switches	Added support for ERSPAN on N9324C-SE1U and N9348Y2C6D-SE1U Smart Switches	10.6(2)F	ERSPAN Guidelines and Limitations for N9300 Smart and Nexus 9324C-SE1U Switches, on page 407
SPAN support on N9300 Smart Switches	Added support for SPAN on N9324C-SE1U and N9348Y2C6D-SE1U Smart Switches	10.6(2)F	N9300 Smart Switches, on page 379

Feature	Description	Changed in Release	Where Documented
GOLD support on N9396T12C-SE1	Added support for Generic On-Line Diagnostics (GOLD) on N9396T12C-SE1 switch	10.6(2)F	Guidelines and limitations for online diagnostics, on page 299
GOLD support on N9396Y12C-SE1	Added support for Generic On-Line Diagnostics (GOLD) on N9396Y12C-SE1 switch	10.6(2)F	Guidelines and limitations for online diagnostics, on page 299
Dropped flow notification	Added support to capture dropped packets and export the information on Nexus 9300-SG2 switches	10.6(2)F	Ingress NetFlow, on page 445 Guidelines and Limitations for Egress NetFlow Dropped flow notification, on page 472
LLDP support on N9336C-SE1 switch	Added support for LLDP on N9336C-SE1 switch	10.6(1)F	Guidelines and Limitations, on page 427
PTP support on N9336C-SE1 switch	Added support for PTP on N9336C-SE1 switch	10.6(1)F	Guidelines and limitations for PTP, on page 72
SPAN support on N9336C-SE1 switch	Added support for SPAN on N9336C-SE1 switch	10.6(1)F	Cisco Nexus 9336C-SE1 switch, on page 378
ERSPAN support on N9336C-SE1 switch	Added support for ERSPAN on N9336C-SE1 switch	10.6(1)F	Guidelines and Limitations, on page 401
sFlow support on N9336C-SE1 switch	Added support for sFlow on N9336C-SE1 switch	10.6(1)F	Guidelines and Limitations, on page 496
SPAN for L2 ports on 9800 switches	Added support for SPAN on Layer 2 ports on Nexus 9800 switches	10.6(1)F	Cisco Nexus 9800 Platform Switches , on page 382
ERSPAN for L2 ports on 9800 switches	Added support for ERSPAN Layer 2 ports on 9800 switches	10.6(1)F	Guidelines and Limitations, on page 401
SPAN to CPU support on 9300-SG2 switches	Added support for SPAN to CPU on Nexus 9364E-SG2 switches.	10.6(1)F	Cisco Nexus 9364E-SG2 switches, on page 379
ERSPAN Tx mirroring support on 9300-SG2 switches	Added support for ERSPAN Tx mirroring packets on Nexus 9364E-SG2 switches.	10.6(1)F	Guidelines and Limitations, on page 401

Feature	Description	Changed in Release	Where Documented
Egress Traffic Analytics support on FX and FX2	Added support for egress Traffic Analytics on 9300-FX and -FX2 TOR switches	10.6(1)F	Guidelines and limitations for Traffic Analytics, on page 488
Packet truncation	Added support for packet truncation	10.6(1)F	Packet truncation, on page 529
Support AuditD	Added support for AuditD feature.	10.6(1)F	Configure Audit Log Reporting, on page 179

New and Changed Information



CHAPTER 2

Overview

This chapter describes the system management features that you can use to monitor and manage Cisco NX-OS devices.

- Overview, on page 7

Overview

Cisco NX-OS offers robust network management and monitoring capabilities, supporting various configuration methods such as CLI, XML management interface, and Cisco DCNM. It features advanced diagnostics, automation, security, and monitoring tools to ensure reliable operation across supported Nexus platforms, with comprehensive documentation and licensing requirements. The following section provides a brief introduction into the various features covered in this document.

Licensing Requirements

See the Cisco NX-OS Licensing Guide and Cisco NX-OS Licensing Options Guide for Cisco NX-OS licensing recommendations and instructions to obtain and apply licenses.

Supported Platforms

See the Nexus Switch Platform Support Matrix to know from which Cisco NX-OS releases various Cisco Nexus 9000 and 3000 switches support a selected feature.

Software Image

The Cisco NX-OS software consists of one NXOS software image.

Cisco NX-OS Device Configuration Methods

Cisco NX-OS device configuration methods are the approaches available to configure network devices, either directly or through management services.

- Direct network configuration using CLI via SSH, Telnet, or console port.
- Configuration through Cisco DCNM client.

Available Configuration Methods and References

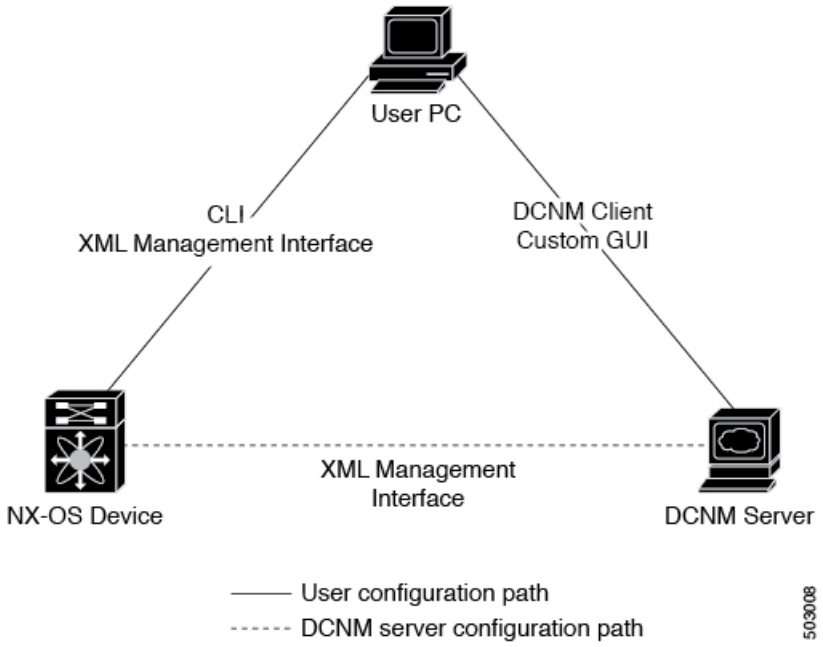
This section lists the configuration methods and the documents where you can find more information.

Table 2: Configuration Methods Book Links

Configuration Method	Document
CLI from a Secure Shell (SSH) session, a Telnet session, or the console port	
Cisco DCNM client	*Cisco DCNM Fundamentals Guide*

This figure shows the device configuration methods available to a network user.

Figure 1: Cisco NX-OS Device Configuration Methods



Configure with CLI or XML Management Interface

You can configure Cisco NX-OS devices using the command-line interface (CLI) or the XML management interface over Secure Shell (SSH) as follows:

- CLI from an SSH session, a Telnet session, or the console port—You can configure devices using the CLI from an SSH session, a Telnet session, or the console port. SSH provides a secure connection to the device. For more information, see the Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide.
- XML management interface over SSH—You can configure devices using the XML management interface, which is a programmatic method based on the NETCONF protocol that complements the CLI functionality. For more information, see the *Cisco NX-OS XML Management Interface User Guide*.

Configure with Cisco DCNM

You can configure Cisco NX-OS devices using the Cisco DCNM client, which runs on your local PC and uses web services on the Cisco DCNM server. The Cisco DCNM server configures the device over the XML

management interface. For more information about the Cisco DCNM client, see the Cisco DCNM Fundamentals Guide.

Network Time Protocol

The Network Time Protocol (NTP) synchronizes the time of day among a set of distributed time servers and clients so that you can correlate time-specific information, such as system logs, received from the devices in your network.

Cisco Discovery Protocol

You can use the Cisco Discovery Protocol (CDP) to discover and view information about all Cisco equipment that is directly attached to your device. CDP runs on all Cisco-manufactured equipment including routers, bridges, access and communication servers, and switches. CDP is media and protocol independent, and gathers the protocol addresses of neighboring devices, discovering the platform of those devices. CDP runs over the data link layer only. Two systems that support different Layer 3 protocols can learn about each other.

Session Manager

Session Manager allows you to create a configuration and apply it in batch mode after the configuration is reviewed and verified for accuracy and completeness.

Scheduler

The scheduler allows you to create and manage jobs such as routinely backing up data or making quality of service (QoS) policy changes. The scheduler can start a job according to your needs—only once at a specified time or at periodic intervals.

Simple Network Management Protocol

The Simple Network Management Protocol (SNMP) is an application-layer protocol that provides a message format for communication between SNMP managers and agents. SNMP provides a standardized framework and a common language used for the monitoring and management of devices in a network.

Online Diagnostics

Cisco Generic Online Diagnostics (GOLD) define a common framework for diagnostic operations across Cisco platforms. The online diagnostic framework specifies the platform-independent fault-detection architecture for centralized and distributed systems, including the common diagnostics CLI and the platform-independent fault-detection procedures for boot-up and run-time diagnostics. The platform-specific diagnostics provide hardware-specific fault-detection tests and allow you to take appropriate corrective action in response to diagnostic test results.

Onboard Failure Logging

You can configure a device to log failure data to persistent storage, which you can retrieve and display for analysis at a later time. This on-board failure logging (OBFL) feature stores failure and environmental information in nonvolatile memory on the module. This information is useful for analysis of failed modules.

Switched Port Analyzer (SPAN)

You can configure an Ethernet Switched Port Analyzer (SPAN) to monitor traffic in and out of your device. The SPAN features allow you to duplicate packets from source ports to destination ports.

Encapsulated Remote Switched Port Analyzer (ERSPAN)

Encapsulated Remote Switched Port Analyzer (ERSPAN) is used to transport mirrored traffic in an IP network. ERSPAN supports source ports, source VLANs, and destinations on different switches, which provide remote monitoring of multiple switches across your network.

To configure an ERSPAN source session, you associate a set of source ports or VLANs with a destination IP address, ERSPAN ID number, and virtual routing and forwarding (VRF) name.

Link Layer Discovery Protocol (LLDP)

Link Layer Discovery Protocol (LLDP) is a vendor-neutral, one-way device discovery protocol that allows network devices to advertise information about themselves to other devices on the network. This protocol runs over the data-link layer, which allows two systems running different network layer protocols to learn about each other. You can enable LLDP globally or per interface.

MPLS Stripping

MPLS stripping provides the ability to strip MPLS labels from packets, enabling non-MPLS-capable network monitoring tools to monitor packets.

Sampled flow (sFlow)

Sampled flow (sFlow) allows you to monitor real-time traffic in data networks that contain switches and routers and to forward the sample data to a central data collector.

Software Maintenance Upgrade (SMU)

A software maintenance upgrade (SMU) is a package file that contains fixes for a specific defect. SMUs are created to respond to immediate issues and do not include new features. SMUs are not an alternative to maintenance releases. They provide a quick resolution of immediate issues. All defects fixed by SMUs are integrated into the maintenance releases.

Virtual Device Contexts

Cisco NX-OS can segment operating system and hardware resources into virtual device contexts (VDCs) that emulate virtual devices. The Cisco Nexus 9000 Series switches currently do not support multiple VDCs. All switch resources are managed in the default VDC.

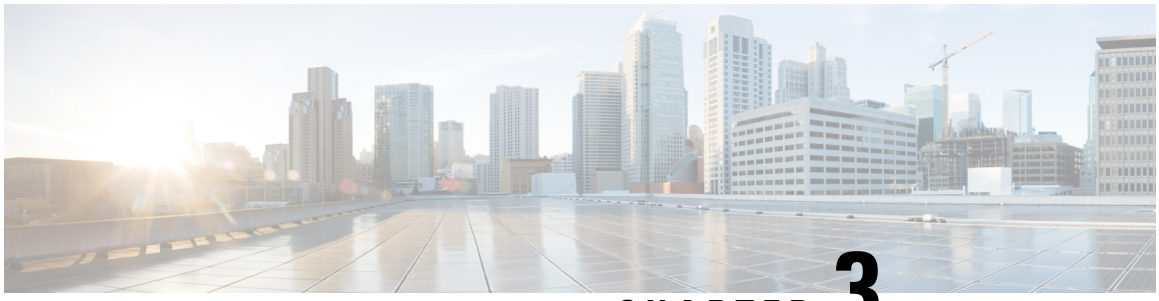
Troubleshooting Features

Cisco NX-OS provides troubleshooting tools such as ping, traceroute, Ethanalyzer, and the Blue Beacon feature.

When a service fails, the system generates information that can be used to determine the cause of the failure. The following sources of information are available:

- Every service restart generates a syslog message of level LOG_ERR.
- If the Smart Call Home service is enabled, every service restart generates a Smart Call Home event.
- If SNMP traps are enabled, the SNMP agent sends a trap when a service is restarted.
- When a service failure occurs on a local module, you can view a log of the event by entering the **show processes log** command in that module. The process logs are persistent across supervisor switchovers and resets.

- When a service fails, a system core image file is generated. You can view recent core images by entering the **show cores** command on the active supervisor. Core files are not persistent across supervisor switchovers and resets, but you can configure the system to export core files to an external server using the file transfer utility Trivial File Transfer Protocol (TFTP) by entering the **system cores** command.
- CISCO-SYSTEM-MIB contains a table for cores (cseSwCoresTable).



CHAPTER 3

Two-stage Configuration Commit

This chapter describes how to enable two-stage configuration commit mode on the Cisco NX-OS device.

This chapter includes the following sections:

- Two-stage configuration commit process, on page 13
- Guidelines and limitations for two-stage configuration commit, on page 14
- Configuring in Two-Stage Configuration Commit Mode, on page 15
- Abort the two-stage configuration commit mode, on page 20
- Display commit IDs, on page 21
- Rollback capability, on page 21
- View current session configurations, on page 22

Two-stage configuration commit process

Configuration Commit is the process of applying configuration changes to a network device. It determines when and how configuration changes take effect on the device. Based on how the configuration is committed, it divided into two modes—one-stage and two-stage.

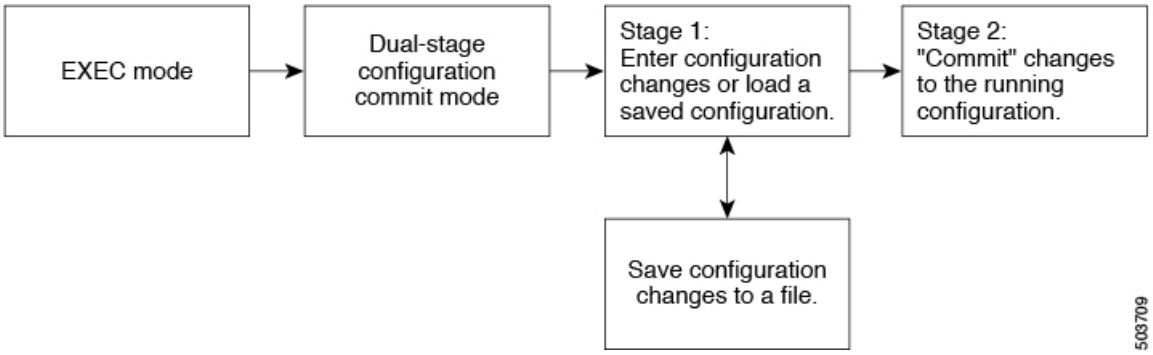
Summary

In an interactive session, when you run a command, it is executed and it changes the running configuration. This is known as one-stage configuration commit. In confirm-commit or two-stage configuration commit mode, changes in configurations are stored in a staging database. These changes do not affect the running configuration until you run the **commit** command.

The two-stage process creates a target configuration session, where you can make, edit, and verify configuration changes before committing them to the running state of the switch. You can also commit the changes for a time period you specify before you commit them permanently. After the specified time period, the switch reverts to the previous configuration if you do not run the **commit** command. When a commit is successful, you can view the commit information that includes the commit ID, username, and timestamp.

Workflow

Figure 2: Two-stage configuration commit process



The image depicts the two stages of the dual-stage configuration commit mode—configuration changes and commit changes.

Guidelines and limitations for two-stage configuration commit

This section lists the configuration guidelines and limitations divided into various categories for two-stage configuration commit.

Before you run any feature-related configuration commands, enable the feature using the **feature** command and commit it using the **commit** command.

Configuration

- This feature is supported only for a command line interface in a user-interactive session.
- Two-stage configuration commit mode does not support other modes like maintenance mode, scheduler mode, or virtual mode.
- When you are in the two-stage configuration commit mode, avoid editing configurations in one-stage configuration commit mode from different sessions at the same time.
- If the verification fails, edit and retry the commit.
- If the commit fails, the configuration rolls back to the previous configuration.
- Configurations that you do not commit are not saved after you reload the switch.
- This feature does not support commits with NX-API, EEM, PPM, and NETCONF.
- You can have only one active two-stage configuration commit session at a given time.

  An SSH connection can terminate unexpectedly without ending the associated two-stage configuration session. When this occurs, the session remains active, and a subsequent `configure dual-stage` command displays `Session is already in use`. Run `show configuration dual-stage sessions` to identify the active session.
- Only users with the network-admin role can use two-stage configuration commit.

Show configuration

- Review the configurations using the **show configuration** command before committing the changes.

- Show configuration displays the staged configs:
  - It displays the real difference, that is yes and no form of the same command will result in empty config.
  - It is recommended to issue the exact no form of the command to negate the configuration.

    For example, to negate **ip address** *x* config, user has to give **no ip address***x* instead of **no ip address**.
  - Interface layer change commands (switchport or no switchport) should be issued explicitly.
  - Any invalid configuration in the session should manually be removed by the user before attempting commit. If could not remove manually clear the session and start a new session.

Platform support

Configuring in Two-Stage Configuration Commit Mode

To enable a feature in the two-stage configuration commit mode, perform the following steps:

Note In this procedure, the BGP feature is enabled as an example.

Procedure

	Command or Action	Purpose

Step 1	**configure dual-stage**
	Example:
	`switch# configure dual-stage`
	`switch(config-dual-stage)#`
		Creates a new target configuration session.
		Note The target configuration isn't a copy of entered during the target configuration

Step 2	**feature** *feature_name*
	Example:
	`switch(config-dual-stage)# feature bgp`
	`switch(config-dual-stage)#`
		Enables the feature.
		Note
		• You can enable the feature even be
		• You can't combine feature-related

Step 3	**commit** [**confirmed** *seconds*]
	Example:
	`switch(config-dual-stage-router)# commit confirmed 30`
	`Verification Succeeded.`
	`Proceeding to apply configuration. This might take a while depending on amount of configuration in`
		Commits changes to the running configuration.
		• **confirmed**: Commits the changes to the running configuration.
		• *seconds*: Commits the configuration in global configuration mode on a trial basis for a minimum of 30 seconds and a maximum of 65535 seconds.

	Command or Action	Purpose
	buffer. Please avoid other configuration changes during this time. Configuration committed by user 'admin' using Commit ID : 1000000001 switch(config-dual-stage)# switch(config-dual-stage)# commit Confirming commit for trial session. switch(config-dual-stage)# **Example:** switch(config-dual-stage)# hostname example-switch switch(config-dual-stage)# commit Verification Succeeded. Proceeding to apply configuration. This might take a while depending on amount of configuration in buffer. Please avoid other configuration changes during this time. Configuration committed by user 'admin' using Commit ID : 1000000002 example-switch(config-dual-stage)#	**Note** If you enter a trial period, run the commit co command, the switch reverts to the previou
Step 4	**Example:** switch(config-dual-stage)# router bgp 64515.46 switch(config-dual-stage-router)# switch(config-dual-stage-router)# router-id 141.8.139.131 switch(config-dual-stage-router)#	Run any feature-related commands that are supported in this configuration mode.
Step 5	**show configuration** **Example:** switch(config-dual-stage-router)# show configuration ! Cached configuration ! router bgp 64515.46 router-id 141.8.139.131	Displays the target configuration. **Note** You can run this command only in the dual
Step 6	**commit** [**confirmed** *seconds*] **Example:** switch(config-dual-stage-router)# commit Verification Succeeded. Proceeding to apply configuration. This might take a while depending on amount of configuration in buffer. Please avoid other configuration changes during this time. Configuration committed by user 'admin' using Commit ID : 1000000003	Commits changes to the running configuration.
Step 7	(Optional) **show configuration commit** [**changes**] *commit-id* **Example:**	Displays commit-related information. Only the last 50 commits or the commit files stored in the reserved disk space are saved. The reserved disk space is

	Command or Action	Purpose
	switch(config-dual-stage-router)# show configuration commit changes 1000000003 *** /bootflash/.dual-stage/1000000003.tmp Fri Mar 19 10:59:00 2021 --- /bootflash/.dual-stage/1000000003 Fri Mar 19 10:59:05 2021 *************** *** 378,383 **** --- 378,385 ---- line console line vty boot nxos bootflash:/nxos64.10.1.1.44.bin + router bgp 64515.46 + router-id 141.8.139.131 xml server timeout 1200 no priority-flow-control override-interface mode off **Example:** switch(config-dual-stage)# show configuration commit 1000000003 feature bgp router bgp 64515.46 router-id 141.8.139.131 . . .	20 MB. All the commit sessions will be removed when you reload the switch. However, the commit IDs are not removed. Also, these commit IDs will not be removed in case of write, erase, and reload. Use the **show configuration commit changes** *commit-id* command to view only the changes in the current session of the commit you specify. Use the **show configuration commit** *commit-id* command to view the complete configurations in the commit that you specify, along with few class-map policies. These class-map policies are not new policies but hidden policies. To view the hidden policies, use the **show run all** command.
Step 8	(Optional) **save configuration** *filename* **Example:** switch(config-dual-stage)# save configuration bootflash:test.cfg	Saves the target configurations to a separate file without committing them to the running configuration. **Note** • You can load the target configurati • You can view the configuration file • Some of the user-specific informat • Configs saved in dual stage mode is **file** <> and not using **#show file** <
Step 9	(Optional) **load** *filename* **Example:** switch (config-dual-stage)# show configuration ! Cached configuration switch (config-dual-stage)# load test.cfg switch (config-dual-stage-router)# show configuration ! Cached configuration ! router bgp 1 switch(config-dual-stage-router)#	Loads a target configuration that you saved. After loading a file, you can modify it or commit it to the running configuration. To save the changes, use the **save configuration** *filename* command. You can load a target configuration that you saved using only the **save configuration** *filename* command.

	Command or Action	Purpose

Step 10	(Optional) **clear configuration**
	Example:
	`switch(config-dual-stage)# show configuration`
	`! Cached configuration`
	`!`
	`router bgp 64515.46`
	`router-id 141.8.139.131`
	`switch (config-dual-stage)# clear configuration`
	`switch (config-dual-stage)# show configuration`
	`! Cached configuration`
	`switch (config-dual-stage)#`
		Clears changes made to the target configuration without terminating the configuration session. It deletes any configuration changes that aren't committed.

Step 11	**end**
	Example:
	`switch(config-dual-stage-if)# end`
	`Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]`
		Exits the global dual stage configuration mode.
		If you end a configuration session without committing the configuration changes, you'll be prompted to save changes, discard changes, or cancel the action:
		• Yes: Commits the configuration changes and exit configuration mode
		• No: Exits the configuration mode without committing the configuration changes
		• Cancel: Remains in configuration mode without committing the configuration changes
		Note
		• If you choose to exit when a confirm c still chose to exit, the trial configuratio
		• If the default session times out before the session. In this case, no warning m

Step 12	**show configuration dual-stage sessions**
	Example:
	`switch(config-dual-stage)# show configuration dual-stage sessions`
	`SNo. Session Line User Date`
	`---- ---------- ---------- ---------- ----------`
	`1 8671-17101913 /dev/ttyS0 admin Wed Feb 17 10:56:00 2021`
	`switch(config-dual-stage)# end`
	`switch# show configuration dual-stage sessions`
	`There are no active dual stage sessions`
	`switch#`
		Before you start a configuration session, you must check if there are other configuration sessions in progress. Only single user is allowed to enter the dual stage configuration mode. Therefore, you need to exit the previous session before starting a new one. There are as many as 32 interactive VSH sessions possible, and the show command displays the PID and line information of the dual stage session.
		Note Dual stage mode will be accessible only af

Step 13	**clear configuration commits diskspace** **Example:**	You can delete the oldest configuration commitIDs by entering the clear configuration commits command. The

	Command or Action	Purpose

	`Southlake-2# clear configuration commits diskspace ?`
	`<1-20971> Number of Kilo Bytes of disk space to free`
	
	`Southlake-2# clear configuration commits diskspace 100`
	`Deleting 7 rollback points from '1000005557' to '1000005563'`
	`101 KB of disk space will be freed.`
	`Continue with deletion (yes/no)? [no] y`
	`Southlake-2#`
		clear configuration commits command must be followed by either the amount of disk space to reclaim or the number of commitIDs to delete. To reclaim disk space from the oldest commitIDs, enter the clear configuration commits command followed by the diskspace keyword and number of kilobytes to reclaim.

Step 14	**clear configuration commits oldest**
	Example:
	`switch(config-dual-stage)# clear configuration commits oldest 10`
	`Deleting 10 rollback points '1000000030' to '1000000039'`
	`125 KB of disk space will be freed.`
	`Continue with deletion (yes/no)? [no] n`
		To delete a specific number of the oldest commitIDs, enter the clear configuration commits command followed by the oldest keyword and number of commitIDs to delete.

Step 15	**Show configuration failed**
	Example:
	`switch(config-dual-stage-if)# commit`
	`Verification Succeeded.`
	
	`Proceeding to apply configuration. This might take a while depending on amount of configuration in buffer.`
	`Please avoid other configuration changes during this time.`
	`Failed to commit one or more configuration items.`
	`Commit Failed, Rolling back ...`
	`switch(config-dual-stage)#`
	`switch(config-dual-stage)# show configuration failed`
	`` `config terminal` ``
	`` `router bgp 100 ` ``
	`` `neighbor 2.2.2.2 ` ``
	`` `bfd ` ``
	`Syntax error while parsing 'bfd '`
	
	`` `neighbor 3.3.3.3 ` ``
	`` `bfd ` ``
	`Syntax error while parsing 'bfd '`
	
	`` `interface port-channel23 ` ``
	`` `bfd ` ``
	`Syntax error while parsing 'bfd '`
	
	`` `end` ``
	`` `end` ``
	`switch(config-dual-stage)#`
		Configuration changes are semantically verified during the commit operation, and begins the actual backend commit once the verification is succeeded. A message appears if one or more configuration entry fails while committing. To display an error message and description for a failed configuration, enter the show configuration failed command. This will display the configuration block that failed in last commit. Configuration block is to preserve the configuration context.

	Command or Action	Purpose

Step 16	**show configuration failed noerrors**
	Example:
	`switch(config-dual-stage)# show configuration failed noerror`
	`router bgp 100`
	`  neighbor 2.2.2.2`
	`    bfd`
	`  neighbor 3.3.3.3`
	`    bfd`
	`interface port-channel23`
	`  bfd`
	`switch(config-dual-stage)#`
		To display only the errored config (without a description) for a failed configuration block, enter the show configuration failed noerrors command.

Step 17	**load configuration failed commit**
	Example:
	`switch(config-dual-stage)# load configuration failed commit`
	`switch(config-dual-stage-if)# sh configuration`
	`! Cached configuration`
	`!`
	`router bgp 100`
	` neighbor 2.2.2.2`
	`    bfd`
	`!`
	`interface port-channel23`
	` bfd`
	`switch(config-dual-stage-if)#`
		If the router displays a verification failure message during commit, the configuration changes are not lost. You can modify the target configuration and commit again. But, if the router displays a configuration failure message (backend error) when you attempt to commit a configuration change, the configuration session will reset. But, while you remain in dual-stage configuration mode, you can reload the failed configuration block into the target configuration, correct the errors, and commit the changes.
		To load a failed configuration, enter the load configuration failed commit command. After recovery, correct and commit the configuration or save it to a file to avoid losing it. Please note that while loading, syntactically wrong configurations will get ignored. You can use ‘show configuration’ to view the target configuration.

Abort the two-stage configuration commit mode

When you abort a configuration session, uncommitted changes are discarded and the configuration session ends.

No warning appears before the configuration changes are deleted.

```
switch(config-dual-stage)# router bgp 1
switch(config-dual-stage-router)# neighbor 192.0.2.4
switch(config-dual-stage-router-neighbor)# remote-as 1
switch(config-dual-stage-router-neighbor)# show configuration
! Cached configuration
!
router bgp 1
neighbor 192.0.2.4
remote-as 1
switch(config-dual-stage-router-neighbor)# show run bgp

!Command: show running-config bgp
!Running configuration last done at: Wed Mar 17 16:17:40 2021
!Time: Wed Mar 17 16:17:55 2021
```

```
version 10.1(2) Bios:version
feature bgp


switch(config-dual-stage-router-neighbor)# abort
switch# show run bgp

!Command: show running-config bgp
!Running configuration last done at: Wed Mar 17 16:18:00 2021
!Time: Wed Mar 17 16:18:04 2021

version 10.1(2) Bios:version
feature bgp

switch#
```

Display commit IDs

At each successful commit, the commit ID is displayed in the syslog. The total number of commit IDs saved in the system depends on the configuration size and the disk space available. However, the maximum number of commit IDs stored at any given time is 50.

Use the **show configuration commit list** command to view information about the last 50 commit IDs. Each entry shows the user who committed configuration changes, the connection used to execute the commit, and commit ID timestamp.

```
switch# show configuration commit list
SNo. Label/ID     User     Line         Client     Time Stamp
~~~~ ~~~~~~~~~~~~ ~~~~~~~~ ~~~~~~~~~~~~ ~~~~~~~~~~ ~~~~~~~~~~~~~~~~~~~~~~~~
1    1000000001   admin    /dev/ttyS0   CLI        Wed Jul 15 15:21:37 2020
2    1000000002   admin    /dev/ttyS0   Rollback   Wed Jul 15 15:22:15 2020
3    1000000003   admin    /dev/pts/0   CLI        Wed Jul 15 15:23:08 2020
4    1000000004   admin    /dev/pts/0   Rollback   Wed Jul 15 15:23:46 2020
```

Rollback capability

You can rollback the configuration to any of the previous successful commits.

Use the **rollback configuration** command to rollback to any of the last 50 commits.

```
switch# rollback configuration to ?
1000000015
1000000016
1000000017

:
:

switch#

Each commit ID acts as a (checkpoint or) rollback point. You can rollback to any given
commit ID. When you roll back the configuration to a specific rollback point, you undo
all configuration changes made during the session identified by the commitID for that
rollback point, and you undo all configuration changes made after that point. The
rollback process rolls back the configuration and commits the rolled-back configuration.
The rollback process also creates a new rollback point (commit ID)so that you can roll
back the configuration to the previous configuration.
```

```
switch(config-dual-stage)# rollback configuration to 1000000002
Rolling back to commitID :1000000002
ADVISORY: Rollback operation started...
Modifying running configuration from another VSH terminal in parallel
is not recommended, as this may lead to Rollback failure.

Configuration committed by rollback using Commit ID : 1000000004
switch(config-dual-stage)#
```

View current session configurations

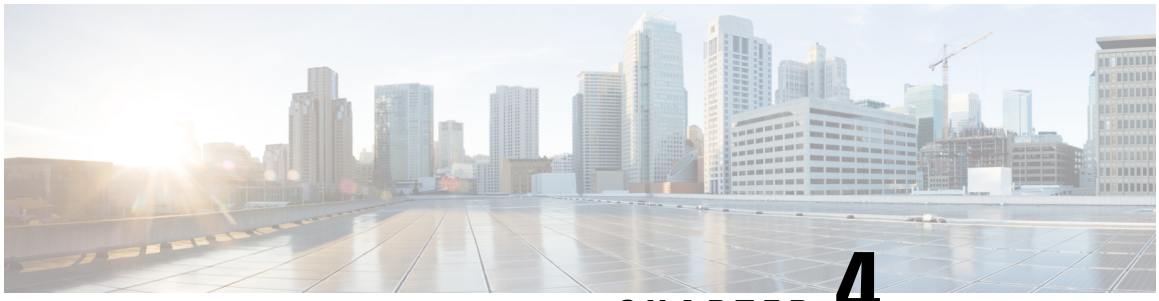
You can view the current session configuration using the **show configuration** command.

This command is supported only in the dual-stage mode. The session configuration is cleared if a commit fails.

```
switch(config-dual-stage-cmap)# show configuration
! Cached configuration
!
class-map type control-plane match-any copp-s-ipmcmiss
class-map type control-plane match-any copp-s-l2switched
class-map type control-plane match-any copp-s-l3destmiss
switch(config-dual-stage-cmap)#

If there is no configuration, the following message appears:

switch(config-dual-stage)# show configuration
! Cached configuration
switch(config-dual-stage)# commit
No configuration changes to commit.
switch(config-dual-stage)#
```



CHAPTER 4

Switch Profiles

- Overview, on page 23
- Guidelines and Limitations, on page 25
- Configure Switch Profile, on page 27
- Add or Modify Switch Profile Commands, on page 29
- Import Switch Profile, on page 30
- Import Configurations in vPC Topology, on page 32
- Isolate Peer Switch, on page 33
- Delete Switch Profile, on page 33
- Manual Correction for Mutex and Merge Failures, on page 34
- Migrate to Cisco NX-OS Release 7.0(3)I2(1) or Higher in a Fabric Extender Straight-Through Topology, on page 34
- Replace Cisco Nexus 9000 Series Switch, on page 35
- Synchronize Configurations After Cisco Nexus 9000 Series Switch Reboots, on page 36
- Synchronize Configurations When mgmt0 Interface Connectivity Is Lost, on page 36
- Revert Inadvertent Port Mode Change of Layer 2 to Layer 3 in Global Configuration Mode, on page 37
- Verify Switch Profile Configuration, on page 37
- Configuration Examples for Switch Profiles, on page 38

Overview

Configuration synchronization (config-sync) is a feature that enables consistent configuration across network devices by automatically synchronizing a switch profile to its peer switch.

- Allows configurations to be synchronized between switches.
- Merges configurations when connectivity is established between two switches.
- Provides control of exactly which configuration gets synchronized.
- Ensures configuration consistency across peers through merge and mutual-exclusion checks.
- Provides verify and commit semantics.
- Allows for migrating existing vPC configurations to a switch profile.

Configuration Modes

The switch profile feature includes the following configuration modes:

- Configuration synchronization mode (config-sync)
- Switch profile mode (config-sync-sp)
- Switch profile import mode (config-sync-sp-import)

Configuration Synchronization Mode

The configuration synchronization mode (config-sync) allows you to create switch profiles.

Switch Profile Mode

The switch profile mode (config-sync-sp) allows you to add supported configuration commands to a switch profile temporary buffer that is later synchronized with a peer switch. Commands that you enter in the switch profile mode are not executed until you enter the **commit** command. Although the syntax of the commands are validated when you enter them, there is no guarantee that the commands will be successful when you enter the **commit** command.

Switch Profile Import Mode

The switch profile import mode (config-sync-sp-import) allows you to import existing switch configurations from the running configuration to a switch profile and specify which commands you want to include in that profile. This option is especially useful when you upgrade from a Cisco NX-OS release that does not support switch profiles to a release that does.

Cisco recommends that you import the necessary configurations from the running configuration using the switch profile import mode and commit the changes before making any additional changes in the switch profile or global configuration mode. Otherwise, you might jeopardize the import, requiring you to abandon the current import session and perform the process again. For more information, see Import Switch Profile, on page 30.

Configuration Validation

Two types of configuration validation checks can identify switch profile failures:

- Mutual exclusion checks
- Merge checks

Mutual Exclusion Checks

The mutual exclusion of configuration commands is enforced in order to avoid duplicate commands in the config-sync and global configuration modes. When you commit the configuration of a switch profile, mutual exclusion (mutex) checks are performed on the local switch as well as the peer switch (if configured). If no failures are reported on both switches, the commit is accepted and pushed into the running configuration.

A command that is included in a switch profile cannot be configured outside of the switch profile.

If a mutex check identifies errors, they are reported as mutex failures, and they must be manually corrected. For details, see Manual Correction for Mutex and Merge Failures, on page 34.

The following exceptions apply to the mutual exclusion policy:

- Interface configuration—An interface configuration can be partially present in a switch profile and partially present in the running configuration as long as there are no conflicts.

- Shutdown/no shutdown
- System QoS

Merge Checks

Merge checks are done on the peer switch that is receiving a configuration. The merge checks ensure that the received configuration does not conflict with the switch profile configuration that already exists on the receiving switch. The merge check occurs during the verify or commit process. Errors are reported as merge failures and must be manually corrected. For details, see Manual Correction for Mutex and Merge Failures, on page 34.

When one or both switches are reloaded and the configurations are synchronized for the first time, the merge check verifies that the switch profile configurations are identical on both switches. Differences in the switch profiles are reported as merge errors and must be manually corrected.

Software Upgrades and Downgrades

You must delete the switch profile when downgrading from a Cisco NX-OS release that supports switch profiles to a release that does not.

When you upgrade from an earlier release to a Cisco NX-OS release that supports switch profiles, you have the option to move some of the running-configuration commands to a switch profile. For details, see Switch Profile Import Mode.

An upgrade can occur if there are buffered (uncommitted) configurations; however, the uncommitted configurations will be lost.

Guidelines and Limitations

Switch profiles have the following configuration guidelines and limitations:

General Guidelines

1. Beginning with Cisco NX-OS Release 9.3(3), the **mtu** command is supported in the interface configuration mode through the switch-profiles configuration mode.
2. You can only enable configuration synchronization using the mgmt0 interface.
3. You must configure synchronized peers with the same switch profile name.
4. Commands that are qualified for a switch profile configuration are allowed to be configured in the configuration switch profile mode (config-sync-sp).
5. Only one switch profile session can be in progress at a time. Attempts to start another session will fail.
6. Command changes made from the global configuration mode are blocked when a switch profile session is in progress.
7. When you enter the **commit** command and a peer switch is reachable, the configuration is applied to both peer switches or neither switch. If a commit failure occurs, the commands remain in the switch profile buffer. You can then make necessary corrections and try the commit again.

Supported Features

- Switch profiles are supported only on Cisco Nexus 9300 Series switches. Cisco Nexus 9500 Series switches do not support switch profiles.
- Supported switch profile commands relate to vPC commands.
- Layer 3 commands are not supported.

Configuration-Synchronization in a Virtual Peer-Link Environment Guidelines

- To initiate a config-sync session with a virtual peer link, be sure to configure a loopback IP address instead of a management IP address between the peer switches.
- You cannot perform a configuration synchronization between a multichassis EtherChannel trunk (MCT) configuration and a virtual peer-link configuration. This config-sync operation is not supported.

Configuration-Synchronization Guidelines

- Port-channels created in the switch profile mode should not be configured using global configuration (config terminal) mode.
- If a port-channel is created in global configuration mode, channel groups including member interfaces must also be created using global configuration mode.
- Port-channels that are configured within the switch profile mode may have members both inside and outside of a switch profile.
- If you want to import a member interface to a switch profile, the port-channel that corresponds with the member interface must also be present within the switch profile.
- For "no system default switchport" configuration at global level, the "switchport" command under port-channel is also considered for mutual exclusion.
- The configuration synchronization (**config-sync**) mode is an L2 mode parallel to the config-terminal mode (**config t**). Config-sync uses the switch-profile to update **config t** mode in the same switch as well as the peer switch. To prevent sync issues in **switch-profile** mode, Cisco recommends that you perform a commit action after each CLI command before overriding, or replacing the current CLI command.

  For example, if you want to overwrite **CLI_command_A** and change it to **CLI_command_B** , commit **CLI_command_A** first, then configure **CLI_command_B** and perform another commit action.

```
switch# conf sync
Enter configuration commands, one per line. End with CNTL/Z.
switch(config-sync)# switch-profile test
Resyncing db before starting Switch-profile.Re-synchronization of switch-profile db
takes a few minutes...
Re-synchronize switch-profile db completed successfully.
Switch-Profile started, Profile ID is 1
switch(config-sync-sp)#
switch(config-sync-sp)# int e 1/3
switch(config-sync-sp-if)# switchport trunk allowed vlan 100-150
switch(config-sync-sp-if)# commit
Verification successful...
Proceeding to apply configuration. This might take a while depending on amount of
configuration in buffer.
Please avoid other configuration changes during this time.
Commit Successful
```

```
switch(config-sync)#
switch(config-sync)# switch-profile test
Resyncing db before starting Switch-profile.Re-synchronization of switch-profile db
takes a few minutes...
Re-synchronize switch-profile db completed successfully.
Switch-Profile started, Profile ID is 1
switch(config-sync-sp)#
switch(config-sync-sp)# int e 1/3
switch(config-sync-sp-if)# switchport trunk allowed vlan 45-90
switch(config-sync-sp-if)# commit
Verification successful...
Proceeding to apply configuration. This might take a while depending on amount of
configuration in buffer.
Please avoid other configuration changes during this time.
Commit Successful
switch(config-sync)# end
switch#
```

Configure Switch Profile

You can create and configure a switch profile on the local switch and then add a second switch that will be included in the synchronization.

You must create the switch profile with the same name on each switch, and the switches must configure each other as a peer. When connectivity is established between switches with the same active switch profile, the switch profiles are synchronized.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal** and enable Cisco Fabric Services (CFS) distribution between the peer switches using the command **cfs ipv4 distribute**

Example:

```
switch# configure terminal
  switch(config)#
```

Example:

```
switch(config)# cfs ipv4 distribute
```

Step 2 Enter the configuration synchronization mode using the command **config sync**

Example:

```
switch(config)# config sync
  switch(config-sync)#
```

Step 3 Configure the switch profile, names the switch profile, and enters the switch profile configuration mode using the command **switch-profile** *name*

Example:

```
switch(config-sync)# switch-profile abc
 switch(config-sync-sp)#
```

Step 4 Add a switch to the switch profile using the command [**no**] **sync-peers destination** *ip-address*

Example:

```
switch(config-sync-sp)# sync-peers destination 10.1.1.1
```

The destination IP address is the IP address of the switch that you want to synchronize.

The **no** form of this command removes the specified switch from the switch profile.



Note You need to wait for peer switches to show the switch-profile status of "In sync" before any commit is done.

Step 5 For Cisco Nexus 3164Q switches only, follow these steps:

a) Enter the switch profile interface configuration mode using the command **interface** *type slot/port*

Example:

```
switch(config-sync-sp)# interface ethernet 1/1
 switch(config-sync-sp-if)#
```

b) Change a Layer 3 interface into a Layer 2 interface using the command **switchport**

Example:

```
switch(config-sync-sp-if)# switchport
```

c) Exit the switch profile interface configuration mode using the command **exit**

Example:

```
switch(config-sync-sp-if)# exit
 switch(config-sync-sp)#
```

d) Commit the current configuration using the command **commit**

Example:

```
switch(config-sync-sp)# commit
```



Note Verify that the switch-profile status shows as "In sync" before any commit is done.

Step 6 Exit the switch profile configuration mode and returns to EXEC mode using the command **end**

Example:

```
switch(config-sync-sp)# end
 switch#
```

This is an optional step.

Step 7 Display the switch profile on the local switch and the peer switch information using the command **show switch-profile** *name* **status**

Example:

```
switch# show switch-profile abc status
```

This is an optional step.

Step 8 Display the switch profile peer configuration using the command **show switch-profile** *name* **peer** *ip-address*

Example:

```
switch# show switch-profile abc peer 10.1.1.1
```

This is an optional step.

Step 9 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch# copy running-config startup-config
```

This is an optional step.

Add or Modify Switch Profile Commands

After you configure a switch profile on the local and the peer switch, you must add and commit the supported commands to the switch profile.

Commands that are added or modified are buffered until you enter the **commit** command. Commands are executed in the same order in which they are buffered. If there is an order dependency for certain commands (for example, a QoS policy must be defined before being applied), you must maintain that order; otherwise, the commit might fail. You can use utility commands, such as the **show switch-profile** *name* **buffer** command, the **buffer-delete** command, and the **buffer-move** command, to change the buffer and correct the order of already entered commands.

Procedure

Step 1 Enter the configuration synchronization mode using the command **config sync**

Example:

```
switch# config sync
 switch(config-sync)#
```

Step 2 Configure the switch profile, name the switch profile, and enter the switch profile configuration mode using the command **switch-profile** *name*

Example:

```
switch(config-sync)# switch-profile abc
 switch(config-sync-sp)#
```

Step 3 Add a command to the switch profile using the command *command*

Example:

```
switch(config-sync-sp)# interface Port-channel100
 switch(config-sync-sp-if)# speed 1000
 switch(config-sync-sp-if)# interface Ethernet1/1
 switch(config-sync-sp-if)# speed 1000
 switch(config-sync-sp-if)# channel-group 100
 switch(config-sync-sp-if)# exit
 switch(config-sync-sp)#
```

Step 4 Display the configuration commands in the switch profile buffer using the command **show switch-profile** *name* **buffer**

Example:

```
switch(config-sync-sp)# show switch-profile abc buffer
```

This is an optional step.

Step 5 Verify the commands in the switch profile buffer using the command **verify**

Example:

```
switch(config-sync-sp)# verify
```

Step 6 Save the commands in the switch profile and synchronize the configuration with the peer switch using the command **commit**

Example:

```
switch(config-sync-sp)# commit
```

This command also does the following:

- Triggers the mutex check and the merge check to verify the synchronization.
- Creates a checkpoint with a rollback infrastructure.
- Executes a rollback on all switches if an application failure occurs on any of the switches in the switch profile.
- Deletes the checkpoint.

Step 7 Exit the switch profile configuration mode and return to EXEC mode using the command **end**

Example:

```
switch(config-sync-sp)# end
  switch#
```

This is an optional step.

Step 8 Display the status of the switch profile on the local switch and the status on the peer switch using the command **show switch-profile** *name* **status**

Example:

```
switch# show switch-profile abc status
```

This is an optional step.

Step 9 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch# copy running-config startup-config
```

This is an optional step.

Import Switch Profile

You can import a switch profile based on the set of commands that you want to import.

Before you begin

Make sure that the switch profile buffer is empty before you import commands to a switch profile.

Procedure

Step 1 Enter configuration synchronization mode and configure the interface that will be imported in Step 4.

Example:

```
switch(config)# interface ethernet 1/2
 switch(config-if)# switchport
 switch(config-if)# switchport mode trunk
 switch(config-if)# switchport trunk allowed vlan 12
 switch(config-if)# speed 10000
 switch(config-if)# spanning-tree port type edge trunk
 switch(config)# end
 switch#
```

This is an optional step.

Step 2 Enter configuration synchronization mode using the command **config sync**

Example:

```
switch# config sync
 switch(config-sync)#
```

Step 3 Configure the switch profile, name the switch profile, and enter the switch profile configuration mode using the command **switch-profile** *name*

Example:

```
switch(config-sync)# switch-profile abc
 switch(config-sync-sp)#
```

Step 4 Identify the commands that you want to import and enter the switch profile import mode using the command **import** [**interface** *interface port* / *slot* | **running-config**]

Example:

```
switch(config-sync-sp)# import interface ethernet 1/2
 switch(config-sync-sp-import)#
```

The following options are available:

- Entering the **import** command without any options adds the selected commands to the switch profile.
- The **import interface** option adds the supported commands for a specified interface.
- The **running-config** option adds supported system-level commands.

Note If new commands are added during the import, the switch profile remains unsaved, and the switch remains in the switch profile import mode.

Step 5 Import the commands and save the commands to the switch profile using the command **commit**

Example:

```
switch(config-sync-sp-import)# commit
```

Step 6 Abort the import process using the command **abort**

Example:

```
switch(config-sync-sp-import)# abort
```

This is an optional step.

Step 7 Exit the switch profile import mode and return to EXEC mode using the command **end**

Example:

```
switch(config-sync-sp-import)# end
 switch#
```

This is an optional step.

Step 8 Display the switch profile configuration using the command **show switch-profile**

Example:

```
switch# show switch-profile
```

This is an optional step.

Step 9 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch# copy running-config startup-config
```

This is an optional step.

Import Configurations in vPC Topology

Importing configurations in a vPC topology is the process of applying configuration profiles to both switches in a two-switch vPC setup to ensure consistent operation.

- Both switches must have a switch profile with the same name.
- Configurations are imported independently to each switch.
- Switches are synchronized using the **sync-peers destination** command.

Steps

Follow these steps to import configurations in a two-switch vPC topology:

1. Configure the switch profile with the same name on both switches.
2. Import the configurations to both switches independently.
3. Configure the switches by entering the **sync-peers destination** command.
4. Verify that the switch profiles are the same by entering the appropriate **show** commands.



Note Make sure that the configuration moved to the switch profile on both switches is identical; otherwise, a merge-check failure might occur.

Isolate Peer Switch

Isolating a peer switch is the process of temporarily removing a switch from its profile to perform configuration changes or troubleshooting without synchronizing with its peer.

Steps

To isolate a peer switch and later restore synchronization, perform the following steps:

1. Remove the peer switch from the switch profile on both switches.
2. Add the **no sync-peers destination** command to the switch profile and commit the changes on both switches.
3. Add any necessary troubleshooting configurations.
4. Verify that the show running switch-profile is identical on both switches.
5. Add the **sync-peers destination** *ip-address* command to both switches and commit the changes.
6. Verify that the peers are in sync.

Note For specific information on the following steps, see the appropriate sections in this chapter.

Delete Switch Profile

You can delete a switch profile.

Procedure

Step 1 Enter configuration synchronization mode using the command **config sync**

Example:

```
switch# config sync
 switch(config-sync)#
```

Step 2 Delete the switch profile using the command **no switch-profile** *name* { **all-config** | **local-config** | **profile-only local** }

Example:

```
switch(config-sync)# no switch-profile abc local-config
   switch(config-sync-sp)#
```

- **all-config** —Deletes the switch profile on the local and the peer switch. If the peer switch is not reachable, only the local switch profile is deleted.
- **local-config** —Deletes the switch profile and local configuration.

- **profile-only local** —Deletes the only the switch profile on the local switch, leaving any configurations previously applied by that profile intact on the local switch.

Note It is recommended that you execute **resync-database** prior to deleting a switch-profile:

```
switch(config-sync)# resync-database
```

Step 3 Exit the switch profile configuration mode and return to EXEC mode using the command **end**

Example:

```
switch(config-sync-sp)# end
 switch#
```

This is an optional step.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch# copy running-config startup-config
```

This is an optional step.

When you enter this command, the config-sync feature triggers the same operation on the peer switch.

Manual Correction for Mutex and Merge Failures

If the conflict is on the peer switch, follow the steps in Isolate Peer Switch, on page 33 to correct the problem on that switch.

1. Import the offending command into the switch profile using the switch profile import mode.
2. Change the behavior as desired.

Migrate to Cisco NX-OS Release 7.0(3)I2(1) or Higher in a Fabric Extender Straight-Through Topology

Migration Steps

The following ordered steps outline the migration process to Cisco NX-OS Release 7.0(3)I2(1) or higher in a Fabric Extender straight-through topology:

1. Make sure configurations are the same on both switches.
2. Configure the switch profile with the same name on both switches.
3. Enter the **import interface port-channel** *x-y* , **port-channel** *z* command for all vPC port channels on both switches.

4. Enter the **show switch-profile** *name* **buffer** command to ensure all configurations are correctly imported on both switches.
5. Remove unwanted configuration settings by editing the buffer.
6. Enter the **commit** command on both switches.
7. Enter the **sync-peers destination** *ip-address* command to configure the peer switch on both switches.
8. Enter the **show switch-profile** *name* **status** command to ensure both switches are synchronized.

Replace Cisco Nexus 9000 Series Switch

This procedure can be done in a hybrid Fabric Extender active/active topology and Fabric Extender straight-through topology.

Note Do not connect any peer link, vPC, active/active, or straight-through topology fabric ports to the replacement switch.

Procedure

Step 1 Boot the replacement switch. The switch comes up with no configuration.

Step 2 Configure the replacement switch.

- If the running configuration was saved offline, follow Steps 4 through 8 to apply the configuration.
- If the running configuration was not saved offline, you can obtain it from the peer switch if the configuration synchronization feature is enabled. (See Steps 1 and 2 in Creating a Switch Profile on a Local and a Peer Switch ; then begin with Step 9 below).
- If neither condition is met, manually add the configuration and then begin with Step 9 below.

Step 3 Edit the configuration file to remove the **sync-peer** command if you are using the configuration synchronization feature.

Step 4 Configure the mgmt port IP address and download the configuration file and copy the saved configuration file to the running configuration.

Step 5 Verify that the configuration is correct by entering the **show running-config** command. If the switch profile configuration changes were made on the peer switch while the replacement switch was out of service, apply those configurations in the switch profile and then enter the **commit** command.

Step 6 Shut down all Fabric Extender straight-through topology ports that are included in a vPC topology and connect the Fabric Extender straight-through topology fabric ports.

Note Wait for the Fabric Extender straight-through topology switches to come online and make sure that the vPC role priority of the existing switch is better than the replacement switch.

Step 7 Connect the peer-link ports to the peer switch and the switch vPC ports.

Step 8 Enter the **no shutdown** command on all Fabric Extender straight-through vPC ports.

Step 9 Verify that all vPC switches and the Fabric Extenders on the replacement switch come online and that there is no disruption in traffic.



Note If you are using the configuration synchronization feature, add the sync-peer configuration to the switch profile if it was not enabled earlier. If you are using the configuration synchronization feature, enter the **show switch-profile** *name* **status** command to ensure both switches are synchronized.

Synchronize Configurations After Cisco Nexus 9000 Series Switch Reboots

Procedure

Step 1 Remove the peer switch from the switch profile on both switches.

Step 2 Add the **no sync-peers destination** command to the switch profile and commit the changes on both switches.

Step 3 Add any missing or changed commands.

Step 4 Verify that the show running switch-profile is identical on both switches.

Step 5 Add the **sync-peers destination** *ip-address* command to both switches and commit the changes.

Step 6 Verify that the peers are in sync.

Synchronize Configurations When mgmt0 Interface Connectivity Is Lost

When the mgmt0 interface connectivity is lost and configuration changes are required, apply the configuration changes on both switches using the switch profile. When connectivity to the mgmt0 interface is restored, both switches are synchronized.

If a configuration change is made on only one switch in this scenario, a merge will succeed when the mgmt0 interface comes up and the configuration gets applied on the other switch.

Revert Inadvertent Port Mode Change of Layer 2 to Layer 3 in Global Configuration Mode

The configurations related to a port imported in config-sync mode should never be configured in the global configuration mode. Normally any attempt to do so will be denied by the config-sync feature, and a mutex warning will appear. However, due to limitations in mutex checks, if a port configured as Layer 2 in the config-sync mode is changed to Layer 3 (no switchport) in the global configuration mode, the config-sync feature is unable to detect and prevent it. As a result, the config-sync mode might become out of sync with the global configuration mode.

Procedure

Step 1 Remove the peer switch from the switch profile on both switches.

Step 2 Add the **no sync-peers destination** command to the switch profile and commit the changes on both switches.

Step 3 Import the current interface configuration.

Step 4 Make any necessary changes and commit them.

Step 5 Verify that the show running switch-profile is identical on both switches.

Step 6 Add the **sync-peers destination** *ip-address* command to both switches and commit the changes.

Step 7 Verify that the peers are in sync.

Verify Switch Profile Configuration

Switch Profile Show Commands

The following table lists the available show commands for switch profiles and describes their purpose.

Table 3: Switch Profile Show Commands

Command	Purpose
show switch-profile *name*	Displays the commands in a switch profile.
show switch-profile *name* **buffer**	Displays the uncommitted commands in a switch profile, the commands that were moved, and the commands that were deleted.
show switch-profile *name* **peer** *ip-address*	Displays the synchronization status for a peer switch.
show switch-profile *name* **session-history**	Displays the status of the last 20 switch profile sessions.
show switch-profile *name* **status**	Displays the configuration synchronization status of a peer switch.

Command	Purpose
show running-config switch-profile	Displays the running configuration for the switch profile on the local switch.
show startup-config switch-profile	Displays the startup configuration for the switch profile on the local switch.

Configuration Examples for Switch Profiles

Display Switch Profile Synchronization Between Local and Peer Switch

The following example shows how to display the initial successful synchronization between the two peers:

```
switch1# show switch-profile sp status

Start-time: 491815 usecs after Mon Jul 20 11:54:51 2015
End-time: 449475 usecs after Mon Jul 20 11:54:58 2015

Profile-Revision: 1
Session-type: Initial-Exchange
Peer-triggered: No
Profile-status: Sync Success

Local information:
----------------
Status: Commit Success
Error(s):

Peer information:
----------------
IP-address: 10.193.194.52
Sync-status: In Sync.
Status: Commit Success
Error(s):


switch2# show switch-profile sp status

Start-time: 503194 usecs after Mon Jul 20 11:54:51 2015
End-time: 532989 usecs after Mon Jul 20 11:54:58 2015

Profile-Revision: 1
Session-type: Initial-Exchange
Peer-triggered: Yes
Profile-status: Sync Success

Local information:
----------------
Status: Commit Success
Error(s):

Peer information:
----------------
IP-address: 10.193.194.51
Sync-status: In Sync.
Status: Commit Success
Error(s):
```

Show Running Configuration

The following example shows the running configuration of the switch profile on the local switch:

```
—— PEER SWITCH-1 ——
switch-1# show running-config switch-profile

switch-profile A
 sync-peers destination 30.0.0.82

 interface port-channel10
 switchport mode trunk
 switchport trunk allowed vlan 10
 spanning-tree port type network
 vpc peer-link

 interface Ethernet2/1
 switchport mode trunk
 switchport trunk allowed vlan 10
 spanning-tree port type network
 channel-group 10 mode active
switch-1#

—— PEER SWITCH-2 ——
switch-2# show running-config switch-profile

switch-profile A
 sync-peers destination 30.0.0.81

 interface port-channel10
 switchport mode trunk
 switchport trunk allowed vlan 10
 spanning-tree port type network
 vpc peer-link

 interface Ethernet2/1
 switchport mode trunk
 switchport trunk allowed vlan 10
 spanning-tree port type network
 channel-group 10 mode active
switch-2#
```

Verify Synchronization Status

The following example shows how to verify the synchronization status between the local and the peer switch:

```
switch-1# show switch-profile status

switch-profile : A
-------------switch-1-----------------------------------------------

Start-time: 912776 usecs after Wed Aug 19 17:03:43 2015
End-time: 868379 usecs after Wed Aug 19 17:03:48 2015

Profile-Revision: 4
Session-type: Commit
Session-subtype: -
Peer-triggered: No
Profile-status: Sync Success

Local information:
----------------
Status: Commit Success
Error(s):
```

```
Peer information:
----------------
IP-address: 30.0.0.82
Sync-status: In sync
Status: Commit Success
Error(s):
```

Create Switch Profile on Local and Peer Switch

The following example shows how to create a successful switch profile configuration on a local and a peer switch, including configuring QoS policies, a vPC peer link, and a vPC in a switch profile.

1. Enable CFS distribution on the local and the peer switch and configure the destination IP address of the switch that you want to synchronize with, such as the management interface on the switch.

```
—Local switch-1#---
switch-1# configure terminal
switch-1(config)# cfs ipv4 distribute
switch-1(config)# interface mgmt 0
switch-1(config-if)# ip address 30.0.0.81/8

—Peer switch-2#--
switch-2# configure terminal
switch-2(config)# cfs ipv4 distribute
switch-2(config)# interface mgmt 0
switch-2(config-if)# ip address 30.0.0.82/8
```

2. Create a new switch profile on the local and the peer switch.

```
—Local switch-1#---
switch-1# config sync
switch-1(config-sync)# switch-profile A
Switch-Profile started, Profile ID is 1
switch-1(config-sync-sp)# sync-peers destination 30.0.0.82
switch-1(config-sync-sp)# end

—Peer switch-2#--
switch-1# config sync
switch-1(config-sync)# switch-profile A
Switch-Profile started, Profile ID is 1
switch-1(config-sync-sp)# sync-peers destination 30.0.0.81
switch-1(config-sync-sp)# end
```

3. Verify that the switch profiles are the same on the local and the peer switch.

```
switch-1(config-sync-sp)# show switch-profile status

switch-profile : A
----------------------------------------------------------

Start-time: 843992 usecs after Wed Aug 19 17:00:01 2015
End-time: 770051 usecs after Wed Aug 19 17:00:03 2015

Profile-Revision: 1
Session-type: Initial-Exchange
Session-subtype: Init-Exchange-All
Peer-triggered: Yes
Profile-status: Sync Success

Local information:
```

```
----------------
Status: Commit Success
Error(s):

Peer information:
----------------
IP-address: 30.0.0.82
Sync-status: In sync
Status: Commit Success
Error(s):
```

4. Add the configuration commands to the switch profile on the local switch. The commands will be applied to the peer switch when the commands are committed.

```
switch-1# config sync
switch-1(config-sync)# switch-profile A
Switch-Profile started, Profile ID is 1
switch-1(config-sync-sp)# interface port-channel 10
switch-1(config-sync-sp-if)# switchport
switch-1(config-sync-sp-if)# commit
Verification successful...
Proceeding to apply configuration. This might take a while depending on amount of
configuration in buffer.
Please avoid other configuration changes during this time.
Commit Successful
switch-1(config-sync)# switch-profile A
Switch-Profile started, Profile ID is 1
switch-1(config-sync-sp)# interface port-channel 10
switch-1(config-sync-sp-if)# switchport mode trunk
switch-1(config-sync-sp-if)# switchport trunk allowed vlan 10
switch-1(config-sync-sp-if)# spanning-tree port type network
switch-1(config-sync-sp-if)# vpc peer-link
switch-1(config-sync-sp-if)# switch-profile switching-mode switchname
switch-1(config-sync-sp-if)# show switch-profile buffer

switch-profile : A
-----------------------------------------------------------
Seq-no Command
-----------------------------------------------------------
1 interface port-channel10
1.1 switchport mode trunk
1.2 switchport trunk allowed vlan 10
1.3 spanning-tree port type network
1.4 vpc peer-link

switch-1(config-sync-sp-if)# commit
Verification successful...
Proceeding to apply configuration. This might take a while depending on amount of
configuration in buffer.
Please avoid other configuration changes during this time.
Commit Successful
switch-1(config-sync)# switch-profile A
Switch-Profile started, Profile ID is 1
switch-1(config-sync-sp)# interface ethernet 2/1
switch-1(config-sync-sp-if)# switchport mode trunk
switch-1(config-sync-sp-if)# switchport trunk allowed vlan 10
switch-1(config-sync-sp-if)# spanning-tree port type network
switch-1(config-sync-sp-if)# channel-group 10 mode active
```

5. View the buffered commands.

```
switch-1(config-sync-sp-if)# show switch-profile buffer

switch-profile : A
-----------------------------------------------------------
Seq-no Command
-----------------------------------------------------------
1 interface Ethernet2/1
1.1 switchport mode trunk
1.2 switchport trunk allowed vlan 10
1.3 spanning-tree port type network
1.4 channel-group 10 mode active
```

6. Verify the commands in the switch profile.

```
switch-1(config-sync-sp-if)# verify
Verification Successful
```

7. Apply the commands to the switch profile and synchronize the configurations between the local and the peer switch.

```
—Local switch-2#--
switch-1(config-sync-sp)# commit
Verification successful...
Proceeding to apply configuration. This might take a while depending on amount of
configuration in buffer.
Please avoid other configuration changes during this time.
Commit Successful
switch-1(config-sync)# end

switch-1# show running-config switch-profile

switch-profile A
sync-peers destination 30.0.0.82

interface port-channel10
switchport mode trunk
switchport trunk allowed vlan 10
spanning-tree port type network
vpc peer-link

interface Ethernet2/1
switchport mode trunk
switchport trunk allowed vlan 10
spanning-tree port type network
channel-group 10 mode active


—Peer switch-2#--
switch-2# show running-config switch-profile

switch-profile A
sync-peers destination 30.0.0.81

interface port-channel10
switchport mode trunk
switchport trunk allowed vlan 10
spanning-tree port type network
vpc peer-link

interface Ethernet2/1
switchport mode trunk
switchport trunk allowed vlan 10
spanning-tree port type network
```

```
    channel-group 10 mode active
```

Display Switch Profile Buffer

The following example shows how to configure the switch profile buffer, the buffer-move configuration, and the buffer-delete configuration:

```
switch1# config sync
switch1(config-sync)# switch-profile sp
Switch-Profile started, Profile ID is 1
switch1(config-sync-sp)# vlan 101
switch1(config-sync-sp-vlan)# ip igmp snooping querier 10.101.1.1
switch1(config-sync-sp-vlan)# exit
switch1(config-sync-sp)# mac address-table static 0000.0000.0001 vlan 101 drop
switch1(config-sync-sp)# interface Ethernet1/2
switch1(config-sync-sp-if)# switchport mode trunk
switch1(config-sync-sp-if)# switchport trunk allowed vlan 101
switch1(config-sync-sp-if)# exit
switch1(config-sync-sp)# show switch-profile sp buffer
-----------------------------------------------------------
Seq-no  Command
-----------------------------------------------------------
1       vlan 101
1.1       ip igmp snooping querier 10.101.1.1
2       mac address-table static 0000.0000.0001 vlan 101 drop
3       interface Ethernet1/2
3.1       switchport mode trunk
3.2       switchport trunk allowed vlan 101

switch1(config-sync-sp)# buffer-move 3 1
switch1(config-sync-sp)# show switch-profile sp buffer
-----------------------------------------------------------
Seq-no  Command
-----------------------------------------------------------
1       interface Ethernet1/2
1.1       switchport mode trunk
1.2       switchport trunk allowed vlan 101
2       vlan 101
2.1       ip igmp snooping querier 10.101.1.1
3       mac address-table static 0000.0000.0001 vlan 101 drop

switch1(config-sync-sp)# buffer-delete 1
switch1(config-sync-sp)# show switch-profile sp buffer
-----------------------------------------------------------
Seq-no  Command
-----------------------------------------------------------
2       vlan 101
2.1       ip igmp snooping querier 10.101.1.1
3       mac address-table static 0000.0000.0001 vlan 101 drop

switch1(config-sync-sp)# buffer-delete all
switch1(config-sync-sp)# show switch-profile sp buffer
```

Display Successful and Unsuccessful Synchronization Between Local and Peer Switch

The following example shows how to configure the synchronization status of the switch profile on the peer switch. The first example shows a successful synchronization, and the second example shows a peer-not-reachable status.

```
switch1# show switch-profile sp peer

switch1# show switch-profile sp peer 10.193.194.52
Peer-sync-status           : In Sync.
Peer-status                : Commit Success
Peer-error(s)              :
switch1#


switch1# show switch-profile sp peer 10.193.194.52
Peer-sync-status           : Not yet merged. pending-merge:1 received_merge:0
Peer-status                : Peer not reachable
Peer-error(s)              :
```

Display 'Verify and Commit' on Local and Peer Switch

The following example shows how to perform a successful verify and commit of the local and the peer switch:

```
switch1# config sync
switch1(config-sync)# switch-profile sp
Switch-Profile started, Profile ID is 1
switch1(config-sync-sp)# interface Ethernet1/1
switch1(config-sync-sp-if)# description foo
switch1(config-sync-sp-if)# exit
switch1(config-sync-sp)# verify
Verification Successful
switch1(config-sync-sp)# commit
Commit Successful
switch1(config-sync)# show running-config switch-profile
switch-profile sp
  sync-peers destination 10.193.194.52
  interface Ethernet1/1
    description foo
switch1(config-sync)# show switch-profile sp status

Start-time: 171513 usecs after Wed Jul 20 17:51:28 2015
End-time: 676451 usecs after Wed Jul 20 17:51:43 2015

Profile-Revision: 3
Session-type: Commit
Peer-triggered: No
Profile-status: Sync Success

Local information:
----------------
Status: Commit Success
Error(s):

Peer information:
----------------
IP-address: 10.193.194.52
Sync-status: In Sync.
Status: Commit Success
Error(s):

switch1(config-sync)#


switch2# show running-config switch-profile
switch-profile sp
  sync-peers destination 10.193.194.51
  interface Ethernet1/1
    description foo
```

```
switch2# show switch-profile sp status

Start-time: 265716 usecs after Mon Jul 20 16:51:28 2015
End-time: 734702 usecs after Mon Jul 20 16:51:43 2015

Profile-Revision: 3
Session-type: Commit
Peer-triggered: Yes
Profile-status: Sync Success

Local information:
----------------
Status: Commit Success
Error(s):

Peer information:
----------------
IP-address: 10.193.194.51
Sync-status: In Sync.
Status: Commit Success
Error(s):
```

Import Configurations

The following example shows how to import an interface configuration:

```
switch# show running-config interface Ethernet1/3

!Command: show running-config interface Ethernet1/3
!Time: Wed Jul 20 18:12:44 2015

version 7.0(3)I2(1)

interface Ethernet1/3
  switchport mode trunk
  switchport trunk allowed vlan 1-100

switch# config sync
switch(config-sync)# switch-profile sp
Switch-Profile started, Profile ID is 1

switch(config-sync-sp)# import interface Ethernet1/3
switch(config-sync-sp-import)# show switch-profile sp buffer
-----------------------------------------------------------
Seq-no  Command
-----------------------------------------------------------
1       interface Ethernet1/3
1.1       switchport mode trunk
1.2       switchport trunk allowed vlan 1-100

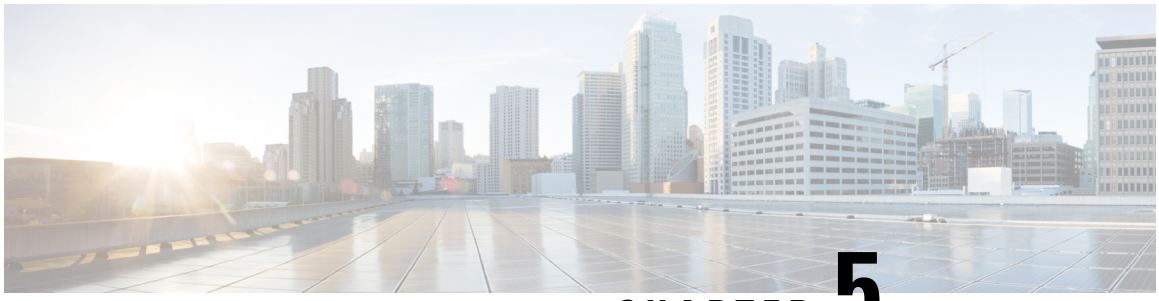
switch(config-sync-sp-import)# verify
Verification Successful
switch(config-sync-sp-import)# commit
Commit Successful
```

The following example shows how to import the supported commands in a running configuration:

```
switch(config-sync)# switch-profile sp
Switch-Profile started, Profile ID is 1
switch(config-sync-sp)# import running-config
switch(config-sync-sp-import)# show switch-profile sp buffer
-----------------------------------------------------------
```

```
Seq-no  Command
-------------------------------------------------------------
1       logging event link-status default
2       vlan 1
3       interface port-channel 3
3.1       switchport mode trunk
3.2       vpc peer-link
3.3       spanning-tree port type network
4       interface port-channel 30
4.1       switchport mode trunk
4.2       vpc 30
4.3       switchport trunk allowed vlan 2-10
5       interface port-channel 31
5.1       switchport mode trunk
5.2       vpc 31
5.3       switchport trunk allowed vlan 11-20
6       interface port-channel 101
6.1       switchport mode fex-fabric
6.2       fex associate 101
7       interface port-channel 102
7.1       switchport mode fex-fabric
7.2       vpc 102
7.3       fex associate 102
8       interface port-channel 103
8.1       switchport mode fex-fabric
8.2       vpc 103
8.3       fex associate 103
9      interface Ethernet1/1
10      interface Ethernet1/2
11      interface Ethernet1/3
12      interface Ethernet1/4
12.1      switchport mode trunk
12.2      channel-group 3
13      interface Ethernet1/5
13.1      switchport mode trunk
13.2      channel-group 3
14      interface Ethernet1/6
14.1      switchport mode trunk
14.2      channel-group 3
15      interface Ethernet1/7
15.1      switchport mode trunk
15.2      channel-group 3
16      interface Ethernet1/8
17      interface Ethernet1/9
17.1      switchport mode trunk
17.2      switchport trunk allowed vlan 11-20
17.3      channel-group 31 mode active
18      interface Ethernet1/10
18.1      switchport mode trunk
18.2      switchport trunk allowed vlan 11-20
18.3      channel-group 31 mode active
19      interface Ethernet1/11
20      interface Ethernet1/12
...
45      interface Ethernet2/4
45.1      fex associate 101
45.2      switchport mode fex-fabric
45.3      channel-group 101
46      interface Ethernet2/5
46.1      fex associate 101
46.2      switchport mode fex-fabric
46.3      channel-group 101
47      interface Ethernet2/6
47.1      fex associate 101
```

```
47.2      switchport mode fex-fabric
47.3      channel-group 101
48      interface Ethernet2/7
48.1      fex associate 101
48.2      switchport mode fex-fabric
48.3      channel-group 101
49      interface Ethernet2/8
49.1      fex associate 101
...
89      interface Ethernet100/1/32
90      interface Ethernet100/1/33
91      interface Ethernet100/1/34
92      interface Ethernet100/1/35
93      interface Ethernet100/1/36
...
105     interface Ethernet100/1/48
```



CHAPTER 5

Configuring Frequency Synchronization

This chapter describes how to configure Frequency Synchronization on Cisco NX-OS devices.

This chapter includes the following sections:

- Frequency synchronization in networks, on page 49
- Licensing requirements for Synchronous Ethernet (SyncE), on page 52
- Guidelines and limitations for Frequency Synchronization, on page 52
- Configure Frequency Synchronization, on page 53

Frequency synchronization in networks

Frequency synchronization is a timing method that

- distributes precision frequency across a network to ensure devices maintain accurate and consistent timing,
- uses Quality Level (QL) information and assigned priority values to select the best available synchronization source, and
- provides precise frequency but does not carry time-of-day information; time-of-day synchronization is achieved separately using packet-based protocols such as Precision Time Protocol (PTP).

Next generation networks must provide the ability to distribute precision frequency around the network. This is known as frequency synchronization. Precision frequency is required for applications such as circuit emulation and cell tower frequency referencing. To comply with ITU specifications for time-division multiplexing (TDM), use differential method circuit emulation. This requires a known, common precision frequency reference at each end of the emulated circuit.

Precisely synchronizing the time-of-day between different network devices is often desirable. This synchronization allows accurate calculation of the packet delay between two nodes in the network.

As SDH and SONET equipment is increasingly replaced by Ethernet equipment, this frequency synchronization ability is required over Ethernet ports. Synchronous Ethernet (SyncE) provides PHY-level frequency distribution of known, common precision frequency references.

To maintain SyncE links, a set of operations messages are required. These messages ensure each node derives timing from the most reliable source, and transfer information about the quality of the timing source used to clock the SyncE link. The ITU Standard G.8264 and its related recommendations document a simple protocol that provides a transport channel for Synchronization Status Messages (SSMs) over Ethernet.

Each timing source has a QL associated that indicates the accuracy of the clock. This QL information is transmitted across the network through SSMs over the Ethernet Synchronization Messaging Channel (ESMC) enabling the devices to identify the best available source for synchronization. Priority values can be assigned to particular timing sources on each switch to define a preferred network synchronization flow and help prevent timing loops. The combination of QL information and user-assigned priority levels enables each switch to choose which timing source to use to clock its SyncE as described in the ITU Standard G.781.

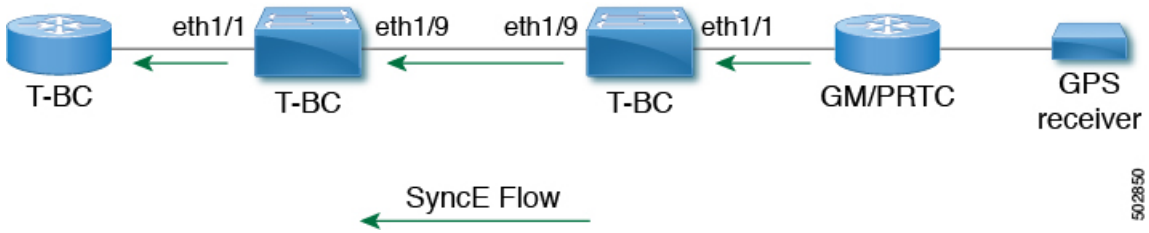
SyncE does not carry time-of-day information. Time-of-day synchronization is achieved using packet-based technologies, such as PTP. Clock sources such as GNSS/GPS can be used to inject accurate time-of-day and frequency into the network. Each switch in the network can select a source for time-of-day as well as a source for frequency (or select the same source for both, if possible and desirable), and pass its time-of-day information to its peers using a packet-based protocol. There is no equivalent to QL for time-of-day information, so selection between different sources for time-of-day is achieved using configuration.

Hybrid SyncE-PTP with external PRC source

The hybrid SyncE-PTP topology helps achieve the end-to-end network precision frequency required for circuit emulation and cell tower frequency referencing.

The figure shows the external timing source as the Grandmaster/Primary Reference Time Clock (GM/PRTC) providing the timing source for the Telecom Boundary Clocks (T-BCs).

Figure 3: Hybrid SyncE-PTP with external PRC source



Timing sources

There are various timing sources that input timing clock signals into the system/network, and outputs of timing clock signals from the system as described below.

Timing inputs

Input clock signals can be received from the platform hardware either via inputs from timing sources like GPS/GNSS, from internal oscillators, recovered from the line of a SyncE enabled interface, or from timing over packet such as the Precision Time Protocol (PTP).

Platform independent (PI) software keeps a database of all these inputs, including a Quality Level (QL) and priority level associated with each. The priority level is configuration controlled, and the QL values can be obtained in a number of ways:

- SyncE enabled interfaces receive SSMs via an Ethernet slow protocol (ESMC).
- GPS and GNSS have fixed QL maintained by platform dependent (PD) software and notified to PI function.
- PTP communicates its QL to the Frequency Synchronization PI software through the platform APIs.

- A default QL value may be defined in the PD layer for the timing connectors, and internal oscillators.
- Configuration may be set defining the QLs of timing sources.

Possible input sources include:

- Internal Oscillator
- Recovered SyncE Clock
- External Clock 1588/PTP
- External Clock (GPS)
- Internal Clock (GNSS)

Timing outputs

The platform hardware can have a number of outputs for clock signals like timing clock outputs from SyncE and enabled interfaces for GPS (currently not supported).

The software keeps all these outputs in a database, including QL information associated with the clock signal being used to drive these outputs that may be explicitly configured. The QL information includes a QL value along with steps removed counters, the originator clock ID, and a series of flags containing information about the path from the originator clock to the current clock. The QL values are transmitted in the same manners as described for the inputs (that is, SyncE interfaces send ESMC SSMs).

Possible output sources:

- SyncE
- IEEE 1588/PTP: packet output is handled separately, in the PTP software.

Timing source selection points

At various stages in syncing timing clocks around the system, the platform has the potential to make a choice over which of the available timing clocks it is to use for further processing. These selection points define the flow of timing clock signals through the system, and eventually lead to the overall decision on which input timing source is to be used for timing outputs.

How these selection points are setup on each platform is hardware dependent, but the platform independent (PI) layer defines a generic selection point abstraction that can flexibly represent any platform selection point hardware, and allows each platform to define which selection points it has, and how they are wired together. The PI code can then control these selection points, tracking and distributing required information about the timing sources, and interacting with the platform dependent (PD) layer to discover what the result of the PD selection is at each stage.

PI timing source selection points include:

- Available Timing Inputs – A number of timing clock inputs are available for the platform selection point hardware to choose between. The availability and associated QL information and priorities are tracked by PI software, which informs the PD layer which inputs are available, ranked in overall order along with their associated quality levels and priorities.
- Platform Specific Selection – The platform layer makes a decision as to which of the inputs it is using based on the information obtained from PI, and other platform layer decisions (for example, hardware level qualification of the clock-signals). The actual decision may be made in PD software (and programmed

into the hardware), or the decision may be made by the hardware itself and communicated back to the PD software.

- Selected Timing Source Outputs – The platform passes the selected clock signal(s) through as output(s) from the selection point. The PD layer informs the PI software the status of the available inputs, and which input(s) have been selected.

The platform layer defines what the selection points are, and how they are connected to potential inputs, and to each other, and to potential outputs. At each of the PD defined selection points, the platform can choose how to interact with the PI software to represent its particular hardware to the PI software. The hardware doesn't have to perform clocking qualification at each selection point. Each selection point simply represents any place where the hardware selects between multiple inputs, passing the clock from one or many inputs forward.

Only one selection point type for SyncE on the switch supervisor is supported. This is named T0 and 1588 selection points. The T0 selection point represents the sources and its selection for the SyncE DPLL. The 1588 selection point represents the sources and its selection for the Assist DPLL for 1588 PLL.

Licensing requirements for Synchronous Ethernet (SyncE)

This table provides information about the document that you can refer for licensing requirements for SyncE.

Product	License Requirement
NX-OS	SyncE requires an add-on license. For a complete explanation of the NX-OS licensing schem NX-OS Licensing Guide .

Guidelines and limitations for Frequency Synchronization

The guidelines and limitations for Frequency Synchronization are:

- SyncE is supported only on physical interfaces.
- A maximum four ethernet interfaces can be monitored for SyncE selection input at any given instance of time.
- Each quad port group on the PHY provides one reference clock.
- Only one Ethernet interface from each quad port group can be configured as a SyncE input (one reference clock for each port group). There is no restriction on SyncE outputs.
- SyncE must be enabled explicitly on the member interfaces for a port-channel. If a member interface of a port-channel is locked as a SyncE source, the ability to send out DNU on other member interfaces enabled for SyncE is controlled via the global command **fsync transmit dnu lag-members** .
- Only G.8275.1 hybrid profile in BC mode is supported.
- For a list of qualified optics for this release, see the Cisco Optics Compatibility Matrix .

Note SyncE is not supported on 1G when GLC-TE is used as SFP.

Platform support

- Refer to Nexus Switch Platform Support Matrix to see the list of Nexus switches that support the Frequency Synchronization (SyncE) feature through NX-OS releases.
- Beginning with NX-OS Release 10.3(2)F, GPS and GNSS is supported on the Nexus 93180YC-FX3 switch.

Configure Frequency Synchronization

Enable Frequency Synchronization

Use this procedure to enable frequency synchronization, set the quality level of the switch, identify the clock ID for ESMC extended TLV, and configure the ESMC peer timeout for software upgrades.

Procedure

Step 1 Enter global configuration mode using the **configure terminal** command.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Enable frequency synchronization on the switch using the [**no**] **feature frequency-synchronization** command.

Example:

```
switch(config)# feature frequency-synchronization
switch(config)#
```

Step 3 Specify the quality level for the switch using the [**no**] **fsync quality itu-t option**{ **1** | **2 generation** { **1** | **2** } command. The default is **option 1**.

Example:

```
switch(config)# fsync quality itu-t option 1
switch(config)#
```

- **option 1** - Includes DNU, EEC1, PRC, PRTC, SEC, SSU-A, SSU-B, eEEC and ePRTC.
- **option 2 generation 1** - Includes DUS, EEC2, PRS, PRTC, RES, SMC, ST2, ST3, ST4, STU, eEEC and ePRTC.
- **option 2 generation 2** - Includes DUS, EEC2, PROV, PRS, PRTC, SMC, ST2, ST3, ST3E, ST4, STU, TNC, eEEC and ePRTC.

Note The quality option that is configured here must match the quality option that is specified in the **quality receive** and **quality transmit** commands in the interface frequency synchronization configuration mode.

Step 4 Specify the clock ID to be used for Ethernet Synchronization Message Channel (ESMC) extended TLV using the **fsync clock-identity** *mac-address* | **no fsync clock-identity** command. If no clock ID is configured, the system uses the default VDC MAC address.

Example:

```
switch(config)# fsync clock-identity AB:CD:EF:12:34:56
switch(config)#
```

Specifies the clock ID to be used for Ethernet Synchronization Message Channel (ESMC) extended TLV. If no clock ID is configured, the system uses the default VDC MAC address.

Step 5 Specify the ESMC peer receive timeout during ISSU using the [**no**] **fsync esmc peer receive timeout**{ **0** | *value*} command.

Zero (0) disables the ESMC peer receive timeout.

value is the ESMC receive timeout in seconds. Enter a value from 120 through 600. Default = 120.

This command ensures that the ESMC control plane, and thus, selection, is not removed during software upgrade for a period of the *value*.

Example:

```
switch(config)# fsync esmc peer receive timeout 120
switch(config)#
```

Step 6 SyncE must be enabled explicitly on the member interfaces for a port-channel using the [**no**] **fsync transmit dnu lag-members** command.

If a member interface of a port-channel is locked as a SyncE source, the ability to send out DNU (Do Not Use) QLs on other member interfaces that are enabled for SyncE is controlled by this command.

If enabled and an interface that is driving the clock for the switch is part of a port-channel, then any members of the port-channel will also send out DNU QL if SyncE is enabled on that interface.

If disabled, the system drives the QL of the selected source on all interfaces regardless of whether they are in the same port-channel as the interface driving the clock.

Example:

```
switch(config)# fsync transmit dnu lag-members
switch(config)#
```

SyncE must be enabled explicitly on the member interfaces for a port-channel. If a member interface of a port-channel is locked as a SyncE source, the ability to send out DNU (Do Not Use) QLs on other member interfaces that are enabled for SyncE is controlled by this command.

If enabled and an interface that is driving the clock for the switch is part of a port-channel, then any members of the port-channel will also send out DNU QL if SyncE is enabled on that interface.

If disabled, the system drives the QL of the selected source on all interfaces regardless of whether they are in the same port-channel as the interface driving the clock.

Step 7 (Optional) Copy the running configuration to the startup configuration using the **copy running-config startup-config** command.

Example:

```
switch(config)# copy running-config startup-config
switch(config)#
```

Copies the running configuration to the startup configuration.

Configure Frequency Synchronization on an interface

Use this procedure to configure frequency synchronization on a specific interface.

This procedure, along with configuring PTP telecom profile on the same interface, constitutes the required interface settings for the hybrid PTP platform. For more information about the interface PTP telecom profile configuration, see Configure PTP Telecom Profile 8275-1 on an Interface , on page 97 .

Before you begin

Make sure that you have globally enabled frequency synchronization on the device (global configuration command **feature frequency-synchronization**).

Procedure

Step 1 Enter global configuration mode using the **configureterminal** command.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Specify the interface on which you are enabling frequency synchronization and enter the interface configuration mode using the [**no**] **interface ethernet***slot* / *port* command.

Example:

```
switch(config)# interface ethernet 1/5
switch(config-if)#
```

Step 3 Enable frequency synchronization on the interface and enter the interface frequency synchronization configuration mode using the [**no**] **frequency synchronization** command.

Example:

```
switch(config-if)# frequency synchronization
switch(config-if-freqsync)#
```

The system selects the frequency signal to be used for clocking transmission, but does not enable the use of the interface as an input.

Note The **no** form of the command functions only if there is no configuration present under the frequency synchronization configuration mode.

Step 4 Specify the interface as a timing source to be passed to the selection algorithm using the [**no**] **selection input** command.

Example:

```
switch(config-if-freqsync)# selection input
switch(config-if-freqsync)#
```

Step 5 Disable sending ESMC packets and ignore any received ESMC packets using the [**no**] **ssm disable** command.

Example:

```
switch(config-if-freqsync)# quality receive exact itu-t option 1 PRC
switch(config-if-freqsync)#
```

Step 6 Adjust the Quality Level (QL) value that is used in received or transmitted SSMs, before it is used in the selection algorithm using the [**no**] **quality** { **receive** | **transmit** } { **exact** | **highest** | **lowest** } **itu-t option***ql-optionql* command.

Each timing source has a QL associated with it which provides the accuracy of the clock. This QL information is transmitted across the network through SSMs over the Ethernet Synchronization Messaging Channel (ESMC) so that devices identify the best available source to use for synchronization.

- **exact***ql* : Specifies the exact QL regardless of the value that is received, unless the received value is DNU.
- **highest***ql* : Specifies an upper limit on the received QL. If the received value is higher than this specified QL, this QL is used instead.
- **lowest***ql* : Specifies a lower limit on the received QL. If the received value is lower than this specified QL, DNU is used instead.

Note The quality option that is specified in this command must match the globally configured quality option in the **quality itu-t option** command.

Example:

```
switch(config-if-freqsync)# quality receive exact itu-t option 1 PRC
switch(config-if-freqsync)#
```

Step 7 Configure the priority of the frequency source on the interface using the [**no**] **priority***value* command.

This priority is used in the clock-selection algorithm to choose between two sources that have the same QL. Values range from 1 (highest priority) to 254 (lowest priority). The default value is 100.

Note This command is valid only if **selection input** is configured.

Example:

```
switch(config-if-freqsync)# priority 100
switch(config-if-freqsync)#
```

Step 8 Configure the wait-to-restore time, in minutes, for frequency synchronization on the interface using the [**no**] **wait-to-restore***minutes* command.

minutes is the amount of time after the interface initializes before it is used for synchronization. Values range from 0 to 12. The default value is 5.

Note This command is valid only if **selection input** is configured.

Example:

```
switch(config-if-freqsync)# wait-to-restore 0
switch(config-if-freqsync)#
```

Commands for verifying frequency synchronization configuration

After performing the frequency synchronization configuration tasks, use this reference to check for configuration errors and verify the configuration.

show frequency synchronization configuration errors

The output of this command displays errors in the frequency synchronization configuration.

The following example shows the mismatch between the global **quality itu-t option** and the interface **quality receive itu-t option** :

```
switch# show frequency synchronization configuration errors

Elysian2(config)# show frequency synchronization configuration errors
Ethernet1/9
    quality receive exact itu-t option 1 PRC
* The QL that is configured is from a different QL option set than is
configured globally.


!Command: show running-config fsync_mgr all
!Running configuration last done at: Mon Feb 10 06:06:15 2020
!Time: Mon Feb 10 06:09:18 2020

version 9.3(5) Bios:version 00.04
feature frequency-synchronization

fsync quality itu-t option 2 generation 1 << must be the same as interface
fsync clock-identity 0
fsync esmc peer receive timeout 120

interface Ethernet1/9
  frequency synchronization
    selection input
    ssm disable
    quality receive exact itu-t option 1 PRC << must be the same as global
    priority 100
    wait-to-restore 0

interface Ethernet1/13
  frequency synchronization
    selection input
    ssm disable
    quality receive exact itu-t option 1 PRC
    priority 110
    wait-to-restore 0
```

show running-config fsync_mgr

The output of this command displays the current frequency synchronization configuration on the device.

The following is an example of the output of the **show running-config fsync_mgr** command:

```
switch# show running-config fsync_mgr

!Command: show running-config fsync_mgr
!Running configuration last done at: Mon Jun 29 13:49:34 2020
!Time: Mon Jun 29 13:50:51 2020

version 9.3(5) Bios:version 01.01
feature frequency-synchronization


interface Ethernet1/9
  frequency synchronization
    selection input
    priority 99
    wait-to-restore 0

interface Ethernet1/13
  frequency synchronization
    selection input
    ssm disable
    quality receive exact itu-t option 1 PRC
    wait-to-restore 0
```

show frequency synchronization interface brief

The output of this command displays all interfaces that have frequency synchronization configured. Sources that have been nominated as inputs have `s` in the Flags (Fl) column. Sources that have not been nominated as inputs do not have `s` displayed.

This is an example of the output of the **show frequency synchronization interface brief** command.

```
switch# show frequency synchronization interface brief

Flags:  > - Up                D - Down              S - Assigned for selection
        d - SSM Disabled      x - Peer timed out    i - Init state
        e - SSM Enabled       s - Output squelched
Fl    Interface          QLrcv QLuse Pri QLsnd Output driven by
===== ================== ===== ===== === ===== ========================
>S    Eth1/9             PRC   PRC   100 PRC   Eth1/13
>Sds  Eth1/13            n/a   PRC   100 n/a   Eth1/13
```

show frequency synchronization interface ethernet

The output of this command displays individual (user-selected) interfaces with associated frequency synchronization information.

This is an example of the output of the **show frequency synchronization interface ethernet** *slot* / *port* command.

```
switch# show frequency synchronization interface ethernet 1/9
Interface State:UP
Assigned as input for Selection
  Wait-to-restore time 0 minute(s)
  SSM Enabled
    Peer Up for 00:07:01, last SSM received 0.307s ago
    Peer has come up 4 times and timed out 1 times
    ESMC SSMs        Total  Information     Event    DNU/DUS
      Sent:           1097         1088         9         83
      Received:        823          816         7        155
  Input:
    Up
    Last received QL: PRC
```

```
    Effective QL: PRC, Priority: 100
    Originator clock ID: ffffffffebfa543
    SyncE steps: 1, eSyncE steps: 1
    Not all steps run eSyncE; Chain of extended ESMC data is broken
    Supports frequency
  Output:
    Selected source: Eth1/13
    Selected source QL: PRC
    Effective QL: PRC
    Originator clock ID: ffffffffebfa863
    SyncE steps: 1, eSyncE steps: 1
    Not all steps run eSyncE; Chain of extended ESMC data is broken
  Next selection points:
```

show frequency synchronization selection (with PTP Profile 8275-1)

The output of this command displays the detailed view of the different selection points within the system.

Note This example shows the output when PTP profile 8275-1 is configured.

This is an example of the output of the **show frequency synchronization selection** *slot* / *port* command.

```
switch# show frequency synchronization selection
==============
Selection point: System Clock (T0) Selector (3 inputs, 1 selected)
  Last programmed 18.898s ago, and selection made 8.621s ago
  Next selection points
    Node scoped   :
  Uses frequency selection
  Used for local line interface output
  S  Input                     Last Selection Point         QL  Pri  Status
  == ========================  ========================  =====  ===  ===========
  11 Ethernet1/9               n/a                         PRC   99  Locked
     Ethernet1/13              n/a                         PRC  100  Available
     Internal0[1]              n/a                         SEC  255  Available
==============
Selection point: IEEE 1588 Clock Selector (3 inputs, 1 selected)
  Last programmed 18.898s ago, and selection made 18.626s ago
  Next selection points
    Node scoped   :
  Uses frequency selection
  S  Input                     Last Selection Point         QL  Pri  Status
  == ========================  ========================  =====  ===  ===========
     Ethernet1/9               n/a                         PRC   99  Unmonitored
     Ethernet1/13              n/a                         PRC  100  Unmonitored
  21 Internal0[1]              n/a                         SEC  255  Freerun    <<
```

show frequency synchronization selection (without PTP Profile 8275-1)

The output of this command displays the detailed view of the different selection points within the system.

Note This example shows the output when PTP profile 8275-1 is not configured.

This is an example of the output of the **show frequency synchronization selection** *slot* / *port* command.

```
switch# show frequency synchronization selection==============
Selection point: System Clock (T0) Selector (3 inputs, 1 selected)
  Last programmed 00:03:04 ago, and selection made 00:02:54 ago
  Next selection points
    Node scoped   :
  Uses frequency selection
  Used for local line interface output
  S  Input                    Last Selection Point         QL  Pri  Status
  == ======================== ======================== ===== ===  ===========
  11 Ethernet1/9              n/a                        PRC   99  Locked
     Ethernet1/13             n/a                        PRC  100  Available
     Internal0[1]             n/a                        SEC  255  Available
==============
Selection point: IEEE 1588 Clock Selector (3 inputs, 1 selected)
  Last programmed 00:03:04 ago, and selection made 3.296s ago
  Next selection points
    Node scoped   :
  Uses frequency selection
  S  Input                    Last Selection Point         QL  Pri  Status
  == ======================== ======================== ===== ===  ===========
     Ethernet1/9              n/a                        PRC   99  Unmonitored
     Ethernet1/13             n/a                        PRC  100  Unmonitored
  21 Internal0[1]             n/a                        SEC  255  Holdover   <<
```

show esmc counters all

The output of this command displays counters for sent and received Ethernet Synchronization Message Channel (ESMC) Synchronization Status Messages (SSMs).

This is an example of the output of the **show esmc counters all** command.

```
ESMC Packet Counters of Interface Ethernet1/1:
    ESMC SSMs        Total  Information     Event    DNU/DUS
      Sent:              0            0         0          0
      Received:          0            0         0          0

ESMC Packet Counters of Interface Ethernet1/5:
    ESMC SSMs        Total  Information     Event    DNU/DUS
      Sent:              0            0         0          0
      Received:          0            0         0          0

ESMC Packet Counters of Interface Ethernet1/9:
    ESMC SSMs        Total  Information     Event    DNU/DUS
      Sent:           7685         7683         2          0
      Received:       7688         7682         6         19

-----------------------------------------------------------------
```

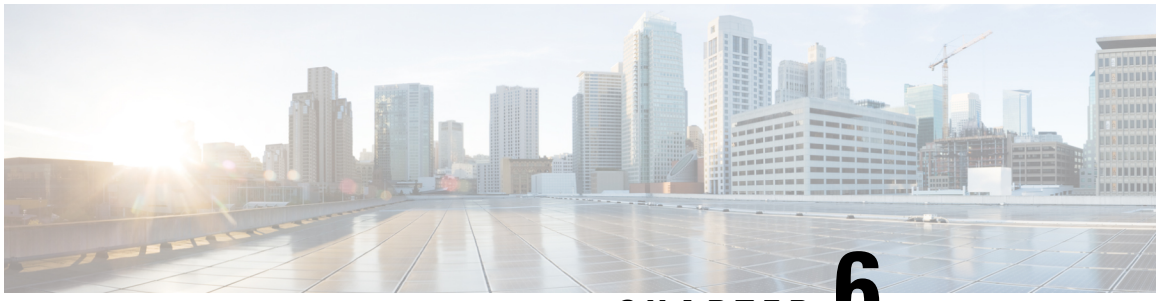
show esmc counters interface ethernet

The output of this command displays counters for sent and received ESMC SSMs on a specific interface.

This is an example of the output of the **show esmc counters interface ethernet** *slot* / *port* command.

```
ESMC Packet Counters of Interface Ethernet1/9:
    ESMC SSMs        Total  Information     Event    DNU/DUS
      Sent:           7955         7953         2          0
      Received:       7958         7952         6         19

-----------------------------------------------------------------
```



CHAPTER 6

Precision Time Protocol

Precision Time Protocol (PTP) is a time synchronization protocol defined in IEEE 1588 for nodes distributed across a network. With PTP, it is possible to synchronize distributed clocks with an accuracy of less than 1 microsecond via Ethernet networks. In addition, PTP's hardware timestamping feature provides timestamp information in the ERSPAN Type III header that can be used to calculate packet latency among edge, aggregate, and core switches.

PTP is a distributed protocol that specifies how real-time PTP clocks in the system synchronize with each other. These clocks are organized into a master-slave synchronization hierarchy with the grandmaster clock, which is the clock at the top of the hierarchy, determining the reference time for the entire system. Synchronization is achieved by exchanging PTP timing messages, with the members using the timing information to adjust their clocks to the time of their master in the hierarchy. PTP operates within a logical scope called a PTP domain.

A PTP system can consist of a combination of PTP and non-PTP devices. PTP devices include ordinary clocks, boundary clocks, and transparent clocks. Non-PTP devices include ordinary network switches, routers, and other infrastructure devices.

- Functionality, on page 61
- Default PTP Parameters, on page 63
- Clocks, on page 64
- Generalized Precision Time Protocol, on page 65
- Process, on page 66
- ITU-T Telecom Profiles, on page 67
- Guidelines and limitations for PTP, on page 72
- Configure PTP, on page 80
- PTP Unicast-Negotiation, on page 113
- Enhanced Multicast Scale, on page 115
- Timestamp Tagging, on page 115
- Related Documents, on page 119

Functionality

PTP supports the following functionality:

- Multicast and unicast PTP transport—In the multicast transport mode, PTP uses multicast destination IP address 224.0.1.129 as per IEEE 1588 standards for communication between devices. For the source IP address, it uses the user configurable global IP address under the PTP domain. In the unicast transport

mode, PTP uses configurable unicast source and destination IP addresses that can be configured under an interface. In both, the unicast and the multicast modes, PTP uses UDP ports, 319 for event messages and 320 for general messages communication between devices.

- PTP multicast configuration is supported only under physical interface for L2 or L3.
- PTP unicast configuration is supported only under physical interface for L2 or L3.
- PTP is not supported with virtual interfaces such as Port-channel, SVI, and tunnel only when configured on associated physical interfaces. When PTP unicast is configured on L2/L3 Port-Channel members, the unicast source IP configuration is derived from the PTP VLAN SVI primary IP or L3 Port-Channel primary IP respectively.
- PTP encapsulation over UDP over IP—PTP uses UDP as the transport protocol over IP. In both, the unicast and multicast modes, PTP uses UDP ports 319 for event messages and 320 for general messages communication between devices. L2 encapsulation mode is not supported.
- PTP profiles—PTP supports default (1588) , AES67, and SMPTE 2059-2 profiles. They all have different ranges of sync and delay request intervals. For information on the default profile, refer to IEEE 1588. For more information on AES67 and SMPTE 2059-2, refer to the respective specifications.
- Path delay measurement—We support delay request and response mechanism to measure the delay between the master and slave devices. Peer delay request and response mechanism is not supported.
- Message intervals—You can configure the interval at which the announce, sync, and delay request messages needs to be sent between devices.
- Best master clock (BMC) selection—BMC algorithm is used to select master, slave, and passive states of the PTP enabled interfaces based on the Announce message received as per 1588 specification.
- PTP Offload - This feature distributes the PTP functionality to the line cards and allows scaling of the number of PTP sessions that are supported on the system. This feature is available for Cisco Nexus 9500 platform switches with N9K-X97160YC-EX line card, 9700-FX, 9636C-R, 9636Q-R, 9624D-R2, and 9636C-RX line cards.
- Dynamic Unicast - This feature allows BMCA to select roles dynamically as follows, rather than assigning static roles.
  - Use the **ptp peer** <ipv4>/<ipv6> command to configure Peer IPs.
  - Port remains in a Listening state until the Peer IP is reachable, when it transitions to Master state.
  - Announce packets are sent as soon as the peer is reachable.
  - Based on Announce packet (using BMCA), Role will be decided. Port state transitions accordingly.
- PTP Time Distribution Hold - In a properly synchronized PTP network, when any PTP node goes down and comes up, the PTP clock is synchronized to its primary time source (GM). During this process, the local node has significant correction and it tries to correct its local clock. At that time, the node can send incorrect time to the downstream nodes and cause issues for all downstream nodes. The Time Distribution (TD) hold feature, introduced in Cisco NX-OS Release 10.5(1)F, resolves this issue by ensuring that the node is properly synchronized to its primary source and distributes time to the downstream nodes during boot up.

  The TD hold feature holds the time distribution until a Boundary Clock (BC) node locks to the primary time source and settles down to the target correction value. The TD hold enabled node receives all PTP packets, does the normal state change, and synchronizes time, but it does not send any PTP packets out.

Note If all nodes reboot at the same time (with a difference of few seconds), each node will be in active hold time, which sometimes results in no nodes having secondary port. This leads to the BMC taking a long time to find the best clock. Hence, the user needs to take this into account when enabling this feature.

- High Availability - Stateful restarts are not supported for PTP. After a reboot or a supervisor switchover, the running configuration is applied. For more information on high availability, see the Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide.
- Mixed Mode - PTP supports Mixed mode for delivering PTP messages, which is detected automatically by Cisco Nexus device, based on the type of **delay_req** message received from connected client and no configuration is required. In this mode when slave sends **delay_req** in unicast message, master also replies with unicast **delay_resp** message.

Default PTP Parameters

The following table lists the default settings for PTP parameters.

Table 4: Default PTP Parameters

Parameters	Default
PTP	Disabled
PTP version	2
PTP domain	0
PTP priority 1 value when advertising the clock	255
PTP priority 2 value when advertising the clock	255
PTP announce interval	1 log second
PTP announce timeout	3 announce intervals
PTP delay-request interval	• 0 log seconds • -1 log seconds for Cisco Nexus 3232C, 3264Q, and 9500 platform switches
PTP sync interval	• -2 log seconds • -3 log seconds for Cisco Nexus 3232C, 3264Q, and 9500 platform switches
PTP VLAN	gPTP supports only default vlan 1, and no other user configured VLANs.

Clocks

Clock Types

The PTP device type is configurable and can be used to set the clock type. The following clocks are common PTP devices:

- Ordinary clock - Communicates with the network based on a single physical port, similar to an end host. An ordinary clock can function as a grandmaster clock.
- Boundary clock - Typically has several physical ports, with each port behaving like a port of an ordinary clock. However, each port shares the local clock, and the clock data sets are common to all ports. Each port decides its individual state, either master (synchronizing other ports connected to it) or slave (synchronizing to a downstream port), based on the best clock available to it through all of the other ports on the boundary clock. Messages related to synchronization and establishing the master-slave hierarchy terminate in the protocol engine of a boundary clock and are not forwarded.
- Transparent clock - Forwards all PTP messages like an ordinary switch or router but measures the residence time of a packet in the switch (the time that the packet takes to traverse the transparent clock) and in some cases the link delay of the ingress port for the packet. The ports have no state because the transparent clock does not need to synchronize to the grandmaster clock. There are two kinds of transparent clocks:
  - End-to-end transparent clock - Measures the residence time of a PTP message and accumulates the times in the correction field of the PTP message or an associated follow-up message.
  - Peer-to-peer transparent clock - Measures the residence time of a PTP message and computes the link delay between each port and a similarly equipped port on another node that shares the link. For a packet, this incoming link delay is added to the residence time in the correction field of the PTP message or an associated follow-up message.
- Grandmaster clock - In a single PTP domain, the Grandmaster (GM) node acts as a primary clock source for the entire PTP network. The primary source for GM node will be taken either from internal GNSS system or external GNSS system. GM node can’t synchronize time or frequency from other PTP node i.e. GM node can’t have any slave ports, all ports act as only Master role.



Note PTP operates only in boundary clock mode. Cisco recommends deployment of a Grand Master Clock (10 MHz) upstream, with servers containing clocks requiring synchronization connected to the switch. End-to-end transparent clock and peer-to-peer transparent clock modes are not supported.

Clock Modes

The IEEE 1588 standard specifies two clock modes for the PTP supporting devices to operate in: one-step and two-step.

One-Step Mode:

In one-step mode the clock synchronization messages include the time at which the master port sends the message. The ASIC adds the timestamp to the synchronization message as it leaves the port. The master port

operating in one-step mode is available for Cisco Nexus 9508-FM-R and 9504-FM-R fabric modules and Cisco Nexus 9636C-R, 9636Q-R, 9624D-R2, and 9636C-RX line cards.

The slave port uses the timestamp that comes as part of the synchronization messages.

Two-Step Mode:

In two-step mode the time at which the synchronization message leaves the port is sent in a subsequent follow-up message. This is the default mode.

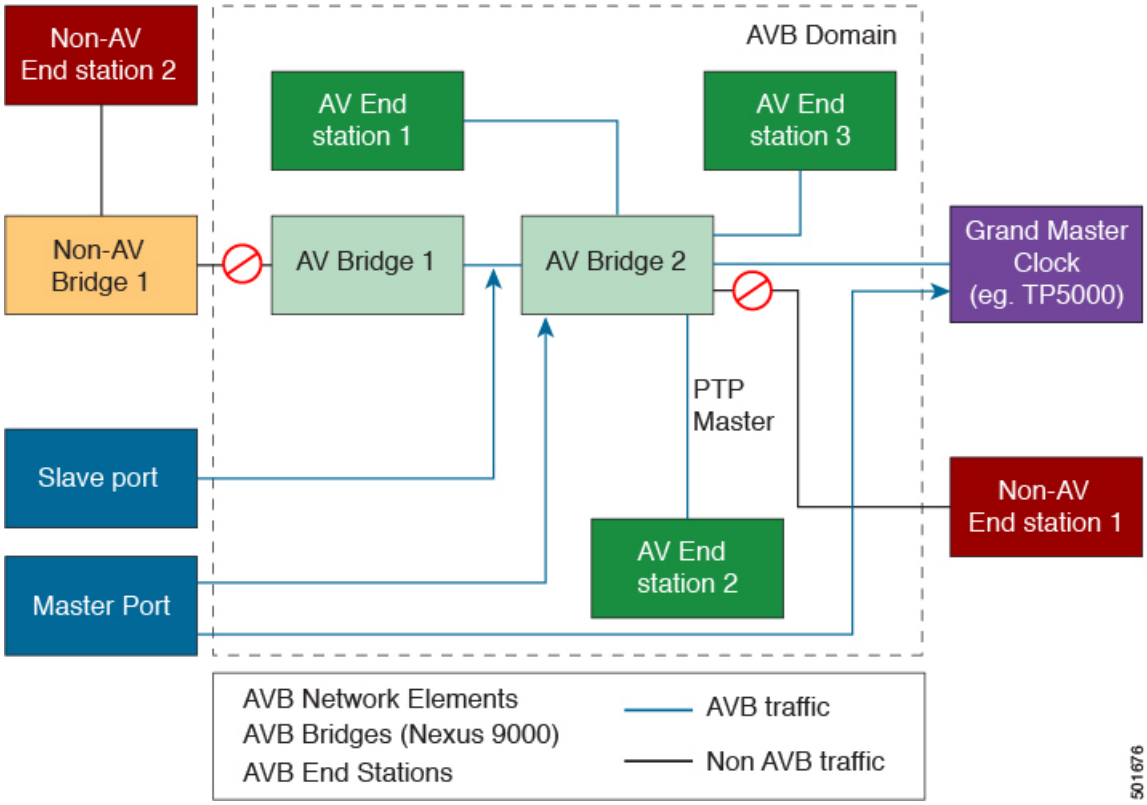
Generalized Precision Time Protocol

Generalized Precision Time Protocol (gPTP) is a time synchronizaion protocol used in audio/video bridging (AVB) networks. It synchronizes endpoints so the listeners correctly use the timestamps carried in the video. gPTP is also required while switching video streams from one source to another on a video frame boundary. gPTP is defined by the IEEE 802.1AS (Timing and Synchronization for Time-Sensitive Applications) standard. Though IEEE 802.1AS is based on IEEE 1588v2, gPTP has the following unique characteristics:

- Mandates the use of only Layer 2 messages.
- Requires that all devices, including the vendor devices, in the network be gPTP compliant.
- Requires the availability of boundary and ordinary clocks for end stations.
- Uses peer delay mechanism two-step clocks and a single gPTP domain.
- Uses the Best Master Clock selection algorithm (BMCA), which behaves as follows:
  - Selects the master/slave state of the port using PTP announce messages.
  - Enables the best clock to be selected as the master.
  - Depends on the quality of the clock, particularly its stability and configuration (such as PTP priority).
- Uses the peer delay mechanism to measure the link delay, which is the delay in propagation of packets from the egress of an interface of the switch at stage N-1 to the ingress of the connected interface of the switch at stage N. It is used between two directly connected devices, which are two ends of the link, using the following peer delay PTP messages: pDelay request, pDelay response, and pDelay response follow-up.
- Synchronizes the clocks of slave devices with the master.

Field	Value
Announce interval 1	1 packet per 2 seconds
Announce receipt timeout	2 seconds
Sync interval 2	1 packet per 4 seconds
Peer delay interval 1	1 packet per 2 seconds

Figure 4: gPTP Master/Slave Relationship



Process

The PTP process consists of two phases: establishing the master-slave hierarchy and synchronizing the clocks.

Within a PTP domain, each port of an ordinary or boundary clock follows this process to determine its state:

- Examines the contents of all received announce messages (issued by ports in the master state)
- Compares the data sets of the foreign master (in the announce message) and the local clock for priority, clock class, accuracy, and so on
- Determines its own state as either master or slave

The ordinary and boundary clocks use **Sync**, **Delay_Req**, **Follow_Up**, **Delay_Resp** event messages to generate and communicate timing information.

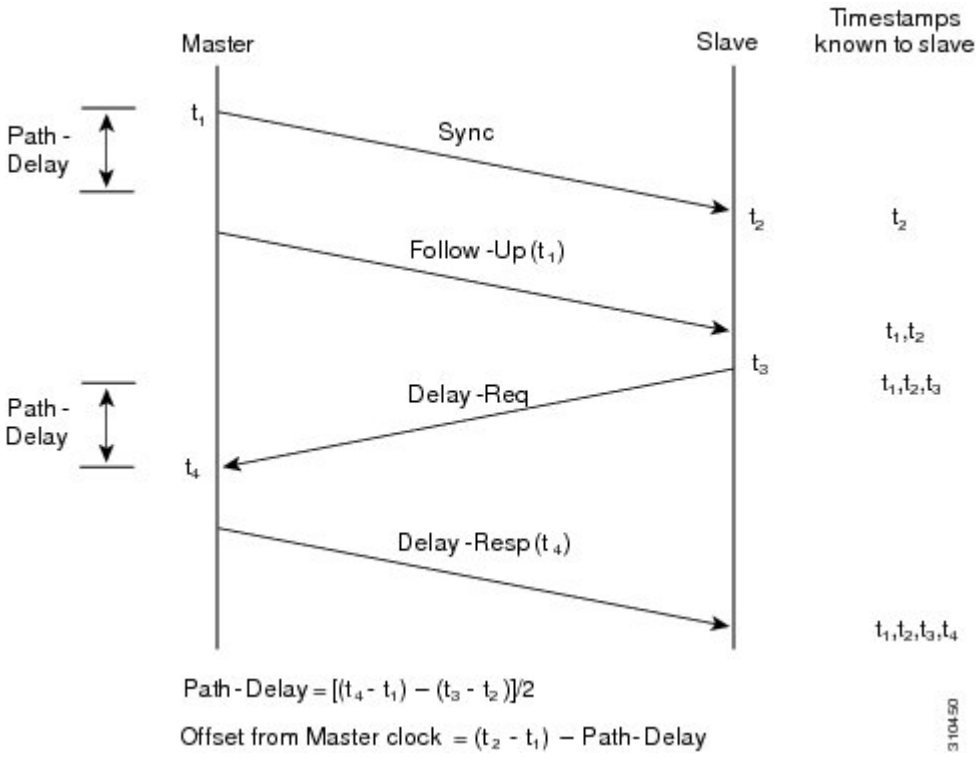
These messages are sent in the following sequence:

1. The master sends a **Sync** message to the slave and notes the time, `t1` at which it was sent. For one-step **Sync** message carries the time when the message leaves the master and for two-step this time is sent in the subsequent **Follow-Up** event message.
2. The slave receives the **Sync** message and notes the time of reception, `t2`.

3. The master conveys to the slave the timestamp, `t1` by embedding the timestamp in a **Follow_Up** event message.
4. The slave sends a **Delay_Req** message to the master and notes the time, `t3` at which it was sent.
5. The master receives the **Delay_Req** message and notes the time of reception, `t4`.
6. The master conveys to the slave the timestamp, `t4` by embedding it in a **Delay_Resp** message.
7. After this sequence, the slave possesses all four timestamps. These timestamps can be used to compute the offset of the slave clock relative to the master, and the mean propagation time of messages between the two clocks.

   The following figure describes the event messages in the PTP process that generate and communicate timing information.

Figure 5: PTP Process



ITU-T Telecom Profiles

Cisco NX-OS software supports ITU-T Telecom Profiles for PTP as defined in the ITU-T recommendation. A profile consists of PTP configuration options applicable only to a specific application.

Separate profiles can be defined to incorporate PTP in different scenarios based on the IEEE 1588-2008 standard. A telecom profile differs in several ways from the default behavior defined in the IEEE 1588-2008 standard and the key differences are mentioned in the subsequent sections.

The following sections describe the ITU-T Telecom Profiles that are supported for PTP:

Profile Defaults

The following table lists the ranges and default values for the commands that are automatically configured when the global command **ptp profile** is set. You cannot change the range for the affected global commands beyond those allowed by the configured profile. However, in the interface mode, they can be changed if the **ptp profile-override** command is set.

Note For Cisco NX-OS Release 9.3(5), only the Cisco Nexus 93180YC-FX3S switch supports either option in this command.

The default profile (default PTP) configurations work with an interval of 4 packets per second (pps). However, on Cisco Nexus 9408, 9808, 9804, N9K-C93400LD-H1, and N9K-C9332D-H2R platform switches, in default and media profile, minimum sync and delay interval supported is 8 pps. When syncing with time source the minimum supported rate for sync and delay req interval is 8 pps.

Table 5: Range and Default Values

Parameter	Scope or Configuration Mode	Default Profile's Supported Range of Values	Default Profile's Default Value	8275-1 Profile's Supported Range of Values	8275-1 Profile's Default Value	8275-2 Profile's Supported Range of Values	8275-2 Profile's Default Value	With 'ptp profile-override' Configured on an Interface Supported Range of Values (Default is Based on Configured Profile)
mode	global	none	none	hybrid	hybrid	non-hybrid	non-hybrid	no change
domain	global	0 to 63	0	24 to 43	24	44 to 63	44	no change
priority1	global	0 to 255	255	128	128	128	128	no change
priority2	global	0 to 255	255	0 to 255	128	0 to 255	128	no change
cost	interface	Not configurable	Not configurable	0 to 255	128	0 to 255	128	0 to 255
transport	interface	ipv4, ipv6	ipv4	ethernet	ethernet	ipv4, ipv6	ipv4	ethernet, ipv4
transmission	interface	multicast, unicast	multicast	multicast	multicast	unicast	unicast	no change
role	interface	dynamic, master, slave	dynamic	dynamic	dynamic	dynamic, master, slave	dynamic	no change

Parameter	Scope or Configuration Mode	Default Profile's Supported Range of Values	Default Profile's Default Value	8275-1 Profile's Supported Range of Values	8275-1 Profile's Default Value	8275-2 Profile's Supported Range of Values	8275-2 Profile's Default Value	With 'ptp profile-override' Configured on an Interface Supported Range of Values (Default is Based on Configured Profile)
announce interval	interface	0 to 4 0 to 4 with aes67 -3 to 1 with smpte-2059-2	1	-3	-3	-3 to 0	-3	-3 to 4 0 to 4 with aes67 -3 to 1 with smpte-2059-2
delay-request minimum interval	interface	-1 to 6 -4 to 5 with aes67 -4 to 5 with smpte-2059-2	0	-4	-4	-4 to 0	-4	-4 to 6 -4 to 5 with aes67 -4 to 5 with smpte-2059-2
sync interval	interface	-3 to 1 -4 to 1 with aes67 -7 to 0 with smpte-2059-2	-2	-4	-4	-4 to 0	-4	-4 to 1 -4 to 1 with aes67 -7 to 0 with smpte-2059-2

Telecom Profile G.8275.1

Cisco's Telecom Profile G.8275.1 feature supports the ITU-T *G.8275.1 : Precision time protocol telecom profile for phase/time synchronization with full timing support from the network* standard. The G.8275.1 profile fulfills the time-of-day and phase synchronization requirements in telecom networks with all network devices participating in the PTP protocol. The G.8275.1 profile with SyncE provides better frequency stability for the time-of-day and phase synchronization.

Features of the G.8275.1 profile are:

- Synchronization Model: G.8275.1 profile adopts hop-by-hop synchronization model. Each network device in the path from master to slave synchronizes its local clock to upstream devices and provides synchronization to downstream devices.

- Clock Selection: G.8275.1 profile also defines an alternate BMCA that selects a clock for synchronization and port states for the local ports of all devices in the network. The parameters defined as a part of the BMCA are:
  - Clock Class
  - Clock Accuracy
  - Offset Scaled Log Variance
  - Priority 2
  - Local Priority
  - Clock Identity
  - Steps Removed
  - Port Identity
- Port State Decision: The port states are selected based on the alternate BMCA.
- Alternate BMCA: It follows the alternate BMCA dataset comparison algorithm as defined in Rec. ITU-T G.8275.1/Y.1369.1 to select the GM for the node.
- Packet Rates: The nominal packet rate for Announce packets is 8 packets-per-second and 16 packets-per-second for Sync/Follow-Up and Delay-Request/Delay-Response packets.
- Transport Mechanism: G.8275.1 profile only supports Ethernet PTP transport mechanism.
- Mode: G.8275.1 profile supports transport of data packets only in multicast mode. The forwarding is done based on forwardable or non-forwardable multicast MAC address.
- Clock Type: G.8275.1 profile supports the following clock types:
  - Telecom Grandmaster (T-GM): Provides timing for other network devices and does not synchronize its local clock to other network devices.
  - Telecom Time Slave Clock (T-TSC): A slave clock synchronizes its local clock to another PTP clock, but does not provide PTP synchronization to any other network devices.
  - Telecom Boundary Clock (T-BC): Synchronizes its local clock to a T-GM or an upstream T-BC clock and provides timing information to downstream T-BC or T-TSC clocks.



Note Telecom Boundary Clock (T-BC) is the only clock type supported in Cisco NX-OS Release 9.3(5).

- Domain Numbers: The domain numbers that can be used in a G.8275.1 profile network ranges from 24 to 43. The default domain number is 24.

Telecom Profile G.8275.2

Cisco's Telecom Profile G.8275.2 feature supports the ITU-T G.8275.2 : *Precision time protocol telecom profile for phase/time synchronization with partial timing support from the network* standard. The G.8275.2

is a PTP profile for use in telecom networks where phase or time-of-day synchronization is required. It differs from G.8275.1 in that it is not required that each device in the network participates in the PTP protocol. Also, G.8275.2 uses PTP over IPv4 and IPv6 in unicast mode.

The G.8275.2 profile is based on the partial timing support from the network. Hence, nodes using G.8275.2 are not required to be directly connected. The G.8275.2 profile is used in mobile cellular systems that require accurate synchronization of time and phase. For example, the fourth generation (4G) of mobile telecommunications technology.

With upcoming technologies such as LTE-TDD, LTE-A CoMP, LTE MBSFN and Location-based services, eNodeBs (base station devices) are required to be accurately synchronized in phase and time. Having GNSS systems at each node is not only expensive, but also introduces vulnerabilities. The G.8275.2 profile meets the synchronization requirements of these new technologies.

PTP Ports

A port can be configured to change its role dynamically. If no role is assigned to a port, it can dynamically assume a primary, passive, or subordinate role based on the BMCA.

In G.8275.2, PTP ports are not tied to any specific physical interfaces, but are tied to a loopback (virtual) interface. Traffic from a PTP port is routed through any physical interface based on the routing decision. For a dynamic port, only one clock source can be configured.

Alternate BPCA

The BPCA (Best Primary Clock Algorithm, which is also known as Best Master Clock Algorithm (BMCA [RFCÂ 7273]) implementation in G.8275.2 is different from that in the default PTP profile. The G.8275.2 implementation specifies an alternate best primary clock algorithm (ABPCA), which is used by each device to select a clock to synchronize to, and to decide the port states of its local ports.

The following consideration apply to the G.8275.2 implementation of the BPCA:

- PrimaryOnly—A per port attribute, PrimaryOnly defines the state of the port. If this attribute is true, the port is never placed in the subordinate state.
- Priority 1—Priority 1 is always static in this profile and is set to 128. Priority 1 is not used in BPCA.
- Priority 2—Priority 2 is a configurable value and its range if from 0 to 255.
- Local Priority—Local priority is configured locally on clock ports to set the priority on nominated clocks. The default value is 128 and valid range is from 1 to 255.

Restrictions for Using the G.8275.2 Profile

- In the G.8275.2 profile, PTP is supported only in the none mode (default).
- A G.8275.2 PTP clock can have redundant clock sources configured (through multiple PTP ports). However, at any given time, a G.8275.2 PTP clock synchronizes to only one clock source, which is selected by BMCA.
- The G.8275.2 does not provide any recommendations for performance analysis and network limits for the clocks.

Guidelines and limitations for PTP

Note

- For scale information, see the release-specific *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*.
- For PTP to function properly, you must use the latest SUP and line card FPGA versions. For information on upgrading FPGAs, access the Release Notes landing page and go to the FPGA/EPLD Upgrade Release Notes (NX-OS Mode Switches) section and locate the FPGA/EPLD Upgrade Release Notes for your software version. Refer to the Installation Guidelines topic.

General Guidelines

- PTP domain limits to a single domain per network.
- PTP can be enabled on the port-channel member ports.
- All management messages that are received from a slave port are forwarded to all PTP enabled ports. The management messages that are received from a slave port are not handled.
- To match PTP control packets using RACL, enable PIM on the L3 interface.
- Before downgrading from Cisco NX-OS Release 9.3(1) to Release 7.0(3)I7, you must unconfigure PTP offload. PTP offloading is not supported for Cisco Nexus 9000 platform switches on 9636PQ, 9564PX, 9464PX, and 9536PQ line cards for Cisco NX-OS Release 7.0(3)I7.
- PTP-capable ports do not identify PTP packets and do not time-stamp or redirect those packets unless you enable PTP on those ports.
- The Cisco Nexus 9504 and 9508 platform switches with 9636C-R, 9636C-RX, or 9636Q-R line cards, the master ports can operate in one-step mode.
- In a topology where PTP is enabled, when a profile is set on a GrandMaster device, and the redundant GrandMasters are deployed in the network; to change the profile on the GrandMaster, you must first shut down the port that is configured on the GrandMaster to the switches, then change the profile, and then re-enable the port. For example, moving from an AES7 profile to an SMPTE profile or conversely.
- The QoS TCAM region Ingress SUP [ingress-sup] must be set to 768 or higher for PTP IPv6 transport to work.
- There is no CLI profile command for 8275.2. This will be added only when the APTS is supported. The functions for this release work only in default mode.
- Beginning with Cisco NX-OS Release 10.4(3)F, transmit and receive scope for PTP over IPv6 multicast configuration is supported.
- On Cisco N9336C-SE1 switches, ports receive PTP Announce and Sync messages even when Spanning Tree Protocol (STP) blocks those ports. This behavior is expected.

Release Specific Guidelines

- *Cisco NX-OS Release 10.6(3)F*
  - PTP is supported on Cisco N9324C-SE1U and N9348Y2C6D-SE1U Smart switches, and the guidelines and limitations include:

- 8275.1 PTP profile is not supported,
- IEEE 1588v2 is supported,
- media profile is supported, and
- PTP is not supported if MACsec is enabled on the interface.

- Cisco N9364E-SG2-Q and N9364E-SG2-O do not support PTP configuration with NETCONF replace.

- *Cisco NX-OS Release 10.6(2)F*
  - PTP is not supported on Cisco Nexus 9336C-SE1 switches if MACsec is enabled on the interface.

- *Cisco NX-OS Release 10.6(1)F*
  - PTP is supported on Cisco Nexus 9336C-SE1 ToR switch, and the guidelines and limitations include:
    - 8275.1 PTP profile is not supported,
    - IEEE 1588v2 is supported, and
    - media profile is supported.

- *Cisco NX-OS Release 10.5(1)F*
  - The PTP Time Distribution (TD) hold feature is introduced. This feature allows for holding the time distribution until a Boundary Clock node locks to the primary time source and settles down to the target correction value.
  - The following attributes are added to the PTP high-correction notification:
    - lastHighCorrectionMPD
    - maxHighCorrectionTime
    - maxHighCorrectionValue
    - maxHighCorrectionMPD

- *Cisco NX-OS Release 10.4(3)F*
  - New global and interface commands are introduced to set the transmit and receive scope for PTP over IPv6 multicast configuration. If upgrading the devices to 10.4(3)F or higher, peer devices running NX-OS should also be upgraded to 10.4(3)F or higher for PTP over IPv6 multicast to work.



Note If any devices are not upgraded, then configure the workaround CLI **test system internal ptp ipv6-mcast-ipv4-dmac int-name** in the devices which are upgraded to 10.4(3)F.

  - The following PTP profiles are supported on Cisco Nexus 9364C-H1 switch:
    - Default profile

- Media profiles (AES67 and SMPTE 2059-2)
- PTP telecom profile (only 8275-2)

- *Cisco NX-OS Release 10.4(2)F*
  - 8275.2 PTP profile is supported on Cisco Nexus 9300-FX3, 9332D-H2R, 93400LD-H1, and 9408 platform switches. The guidelines for this profile are as follows:
    - Supports Unicast negotiation (simple unicast is not supported)
    - Default role for a dynamic unicast port is dynamic. The negotiation is triggered by state transitions. Only one PTP peer can be configured
    - Introduced a new command - **ptp peer** <ipv4>/<ipv6> - to configure a ptp peer when the role is dynamic.
    - For information about DME, see Cisco Nexus NX-API References.
  - PTP Class C is now supported on N9K-C93400LD-H1 switches. The guidelines and limitations for this platform switches are as follows:
    - Only One-Step Mode is supported.
    - System can take a maximum of 30 minutes to achieve Class C level corrections
    - Both G.8275.1 and G.8275.2 Telecom profiles are supported, However, only Telecom Boundary Clock (T-BC) mode is supported.
  - Default PTP is supported on Cisco Nexus C93108TC-FX3 switch.
- *Cisco NX-OS Release 10.4(1)F*
  - PTP is supported on the Cisco Nexus 9804 switches and the following guidelines and limitations apply:
    - Only IPv4 transport is supported.
    - Only media profile is supported.
    - Only supported clock distribution mode is one step. Two step mode is not supported.
    - Other PTP features such as unicast and unicast negotiation are not supported.
  - Cisco Nexus X98900CD-A and X9836DM-A line cards with Cisco Nexus 9808 and 9804 switches
  - PTP is not supported on Cisco Nexus C9348GC-FX3PH switch.
  - PTP is supported on Cisco Nexus C9348GC-FX3 switch.

Note Make sure that you increase COPP limit to have PTP state stable across PTP interfaces. For more information on configuration, see *CoPP Configuration Example* section in *Cisco Nexus 9000 Series NX-OS Security Configuration Guide.*

- ASIC-based time stamping on 1G PTP enabled ports is supported on N9K-C931806C-FX3 and N9K-C93108TC-FX3P platform switches. Thus, both 1G and 10G ports can run on ASIC mode now. However, all the 1G ports must be either in PHY mode (default) or in ASIC mode. A global command is introduced to enable this feature.
- PTP on media profile is supported on N9K-C9332D-H2R platform switch. A few guidelines and limitations for this platform switch are as follows:
  - Only multicast PTP with one-step is supported.
  - Unicast is not supported.
  - PTP is not supported on interface Ethernet 1/33 and Ethernet 1/34.
- PTP v1 packet forwarding feature is supported on Cisco Nexus N9K-C9408 and N9K-C9332D-H2R platform switches.
- The PTP Class C support for G.8275.1 Telecom profile feature is supported on Cisco Nexus 9408 and N9K-C9332D-H2R platform switches. A few guidelines and limitations for this platform switches are as follows:
  - The platform switches provide Class C support
  - Only One-Step Mode is supported
  - Ethernet transport is supported
  - System can take up to 30 minutes to achieve Class C level corrections
  - The G.8275.1 Telecom profile is supported, except the following:
    - Telecom Transparent Clock (T-TC)
    - Telecom Grand Master (T-GM)

- *Cisco NX-OS Release 10.3(5)M*
  - New global and interface commands are introduced to set the transmit and receive scope for PTP over IPv6 multicast configuration. If upgrading the devices to 10.3(5)M or higher, peer devices running NX-OS should also be upgraded to 10.3(5)M or higher for PTP over IPv6 multicast to work.



Note If any devices are not upgraded, then configure the workaround CLI **test system internal ptp ipv6-mcast-ipv4-dmac int-name** in the devices which are upgraded to 10.3(5)M.

- *Cisco NX-OS Release 10.3(3)F*
  - 8275.2 Telecom profile is supported on Cisco Nexus 9300-FX3 platform switches. The following guidelines and limitations apply:
    - This feature allows ITU-T precision time protocol (PTP) profile for phase/time distribution with partial timing support from the network (unicast mode).
    - A command for G.8275.2 is added as the APTS is supported.

- This feature also supports multiple unicast followers. Among the unicast followers, one is picked as an active unicast follower, while the rest configured unicast followers act as passive.
- From this release onwards, dynamic role with unicast ports is supported.
- For dynamic PTP unicast ports, the unicast-negotiation feature must be enabled.

- *Cisco NX-OS Release 10.3(2)F*
  - PTP GM functionality is supported on the Cisco Nexus N9K-C93180YC-FX3 platform switches.
  - PTP on media profile is supported on Cisco Nexus 9408 platform switches. A few guidelines and limitations for this platform switches are as follows:
    - Only multicast PTP with one-step is supported.
    - Other PTP features such as unicast and unicast negotiation are not supported.

- *Cisco NX-OS Release 10.3(1)F*
  - PTP Media Profile and One-Step Mode are supported on the Cisco Nexus 9808 platform switches. A few guidelines and limitations for this platform switches are as follows:
    - Only IPv4 transport is supported.
    - Only multicast PTP with one step is supported.
    - Other PTP features such as unicast and unicast negotiation are not supported.

- *Cisco NX-OS Release 10.2(3)F*
  - The PTP Support of up to 2000 Secondary Devices per Switch feature provides an option to support a maximum of 100 multicast secondary devices per port, with a system-wide support for a maximum of 2000 multicast secondary devices per switch. This feature is only supported on Cisco Nexus 9000-FX2 and 9000-FX3 platform switches.
  - N9K-C9364D-GX2A - PTP is not supported on 1/65 and 1/66 ports
  - N9K-C9348D-GX2A - PTP is not supported on 1/49 and 1/50 ports

- *Cisco NX-OS Release 10.2(2)F*
  - The PTPv1 and v2 co-existence feature is supported on the Cisco Nexus 9300-GX, 9300-GX2, and 9300-FX3 platform switches.
  - The PTP with jitter fix for 1G ports feature is supported on the Cisco N9K-C93108TC-FX3P platform switches.
  - The PTP IPv6 UDP transport feature is supported on the Cisco Nexus 9300-FX, 9300-FX2, 9300-GX, and 9300-GX2 platform switches.

- *Cisco NX-OS Release 10.2(1)F*
  - Unicast-negotiation is supported for IPv4 and IPv6 addresses with default profile on N9K-C93180YC-FX3S platform.
  - PTP IPv6 transport is supported on N9K-C93180YC-FX3S platform.

- PTP is supported on the N9K-C9332D-GX2B platform switches. However, PTP is not supported on 1/33 and 1/34 ports.

- *Cisco NX-OS Release 10.1(2)F*
  - PTP is supported on the N9K-X9624D-R2 line cards.
  - PTP (IEEE 1588) is supported on the N9K-C9700-GX line card, as well as N9K-X97160YC-EX line card and N9K-C9700-FX line cards, used with N9K-C9504-FM-G and N9K-C9508-FM-G fabric modules.
  - PTP (IEEE 1588) is supported on the N9K-C9504-FM-G and N9K-C9508-FM-G platform switches.
  - Explicit carving of ing-sup (size of the ingress supervisor TCAM region) to 768 is not required for the PTP profile 8275-1.

- *Cisco NX-OS Release 9.3(7)F*
  - PTP G.8275.1 Telecom profile is supported on N9K-C93180YC-FX3 platform switch.

- *Cisco NX-OS Release 9.3(5)F*
  - CLI behavior for PTP commands has changed as follows:
    - Most PTP commands do not return errors if the same command is applied again.
    - Most PTP commands do not validate the parameters that are entered as "no" commands. For example, if the currently configured command is "ptp sync interval -3", "no ptp sync interval -1" is accepted for negation.
  - PTP is supported on Cisco Nexus 9300-GX platform switches.
  - PTP G.8275.1 Telecom profile is supported on N9K-C93180YC-FX3S platform switch.
  - PTP is supported on N9K-C93108TC-FX3P platform switch. However, syncE is not supported.

- *Cisco NX-OS Release 9.3(3)F*
  - PTP is supported on Cisco Nexus 93360YC-FX2 and 93216TC-FX2 switches.

- *Cisco NX-OS Release 9.3(2)F*
  - For Cisco Nexus 9636C-R, 9636Q-R, 9636C-RX, and 96136YC-R line cards, the following PTP configuration commands are defaults:
    - **ptp offload**
    - **ptp clock-operation one-step**

    These default commands enable PTP to be distributed on the line cards and enable all PTP configured ports to run in one-step. We recommend using the default configuration for performance and scalability.

Note Changing the **ptp clock-operation** command to two step will disable the offload and PTP will then run centralized on the supervisor card.

- PTP offload is automatically enabled when one-step mode is configured.

- *Cisco NX-OS Release 9.2(3)F*
  - PTP is available for Cisco Nexus 9504-FM-R platform switches.

- *Cisco NX-OS Release 9.2(1)F*
  - Cisco Nexus 9732C-EX, 9736C-EX, and 97160YC-EX line cards support PTP offloading.

Telecom Profile Guidelines

- PTP Telecom Profile is supported only on the Cisco Nexus 93180YC-FX3S switch and N9K-C93180YC-FX3 switch.
- Only PTP class B is supported for 25G and above port speed.
- Only Telecom Boundary Clock (T-BC) is supported.
- Cisco's Telecom Profile G.8273.2 feature is compliant with the ITU-T *G.8273.2 : Timing characteristics of telecom boundary clocks and telecom time slave clocks* standard with the exception that 1PPS output is not aligned with PTP.



Note Time of Day and PTP GM are not supported in Cisco NX-OS Release 9.3(5).

- Beginning with Cisco NX-OS Release 10.4(3)F, PTP Telecom Profile over MACsec is supported on Cisco Nexus 9300-H2R/H1 platform switches.

Timestamp Tagging (TTAG) Support for Platform Switches

- Cisco Nexus 9200 platform switches: beginning with Cisco NX-OS Release 7.0(3)I6(1)
- Cisco Nexus 9300-FX platform switches: beginning with Cisco NX-OS Release 7.0(3)I7(3)
- Cisco Nexus 9300-FX2 platform switches: beginning with Cisco NX-OS Release 9.3(3)
- Cisco Nexus 9300-FX3 and -GX platform switches: beginning with Cisco NX-OS Release 9.3(5)
- Cisco Nexus 9500 platform switches with -EX and -FX line cards

PTP Support

- PTP transport over User Datagram Protocol (UDP) is supported. PTP over Ethernet is supported only on Cisco Nexus 9300-FX3 platform switches.
- PTP supports the multicast communication. PTP also supports the unicast communication and the unicast mode is optional.
- PTP supports boundary clock mode.
- The PTP correction-range, PTP correction-range logging, and PTP mean-path-delay commands are supported on the Cisco Nexus 9508-R line cards.
- Cisco Nexus 9000 series switches support mixed non-negotiated mode of operation on master PTP ports. That means when a slave client sends unicast delay request PTP packet, the Cisco Nexus 9000 responds

with a unicast delay response packet. And, if the slave client sends multicast delay request PTP packet, the Cisco Nexus 9000 responds with a multicast delay response packet. For mixed non-negotiated mode to work, the source IP address used in the ptp source *IP address* configuration on the BC device must also be configured on any physical or logical interface of the BC device. The recommended best practice is to use the loopback interface of the device.

- The Cisco Nexus 9396 switch supports PTP mixed mode.
- PTP is supported with sync interval -3 only on Cisco Nexus 9508-R family line cards. Higher sync intervals are not supported.
- PTP unicast is supported only on the default vrf.
- PTP supports mixed mode for delivering PTP messages, which is detected automatically by a Cisco Nexus device based on the type of delay request message that is received from a connected client and no configuration is required.
- PTP supports unicast communication on Cisco Nexus 9508 switches with 9636C-R, 9636C-RX, and 9636Q-R line cards.
- PTP one-step mode is supported only in PTP offload mode for the Cisco Nexus 9504 and 9508 platform switches with 9636C-R, 9636C-RX, 9624D-R2, or 9636Q-R line cards.
- Each port can be individually configured with any of the supported PTP profiles. Different PTP profiles can coexist on an interface. Combination of the default of 1588 and SMPTE-2059-2 or AES67 profiles is supported.
- The platform switches are supported only on Class B and do not meet Class C support.
- One-step PTP is supported on the slave port of Cloudscale Nexus 9000 series switches, but not on master ports. Nexus 9000-R series switches support PTP one-step on all ports.
- When running PTP, without NTP, **clock protocol ptp** needs to be configured.

PTP Not Supported

- PTPv1 forwarding and feature VMCT are not supported if enabled at the same time.
- 1 Pulse per Second (1PPS) output is enabled by default. UTC/SMB port is in output mode. Note that 1PPS output is not supported.
- End-to-end transparent clock and peer-to-peer transparent clock modes are not supported.
- For Cisco Nexus 31108PC-V and 31108TC-V switches, PTP is not supported on ports running at 100G speed.
- PTP is not supported for stateful high availability.
- PTP is not supported for management interfaces.
- PTP unicast is not supported in offload mode.
- PTP configuration with UC and MC on either side is not supported on Cisco Nexus 9508 switches with 9636C-R, 9636C-RX, and 9636Q-R line cards.
- PTP is not supported on FEX interfaces.
- A combination of SMPTE-2059-2 and AES67 profiles is not supported on the same interface.

Configure PTP

Configure PTP Globally

You can enable or disable PTP globally on a device. You can also configure various PTP clock parameters to help determine which clock in the network has the highest priority to be selected as the grandmaster.

Note For PTP to function properly, you must use the latest SUP and LC FPGA versions. For information on upgrading FPGAs, access the Release Notes landing page and go to the FPGA/EPLD Upgrade Release Notes (NX-OS Mode Switches) section and locate the FPGA/EPLD Upgrade Release Notes for your software version. Refer to the Installation Guidelines topic.

Note You must always set the clock protocol PTP vdc1 for the local clock to be updated by the PTP protocol, irrespective of the one-step or the two-step mode. You can verify the configuration using the **show running-config clock_manager** command.

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Enable or disable PTP on the device using the command [**no**] **feature ptp**

Example:

```
switch(config)# feature ptp
```

Note Enabling PTP on the switch does not enable PTP on each interface.

Make sure that only one of these features is configured: dot1x (**feature dot1x**) or NV overlay (**feature nv overlay**). A dynamic ifacl label is reserved when these features are configured. However, only two dynamic ifacl label bits are available. If both of these features are already configured, a dynamic ifacl label will not be available for PTP, and the feature cannot be enabled.

Step 3 (Optional) Disable PTP on the each 1G interface[s] before enabling or disabling global CLI - ptp scale-on-1G using the command [**no**] **ptp scale-on-1G**

Example:

```
switch(config)# ptp scale-on-1G
```

Enables time stamping on ASIC for 1G ports (by default, PHY mode is enabled).

Note If you do not disable, both rollback config and config replace will fail. This is required for both when moving PHY based time stamping to ASIC based time stamping and vice versa. Global CLI is applicable to N9K-C93108TC-FX3P only.

Step 4 Configure the device type as gPTP or boundary clock or ordinary clock grandmaster using the command [**no**] **ptp device-type** [**generalized-ptp** | **boundary-clock** | **ordinary-clock-grandmaster**]

Example:

```
switch(config)# ptp device-type generalized-ptp
```

Note The **generalized-ptp** option is available only for Cisco Nexus 9508 switches with an -R series line card .

Note Beginning with Cisco NX-OS Release 10.3(2)F, the ordinary clock grandmaster is available only for Cisco Nexus N9K-C93180YC-FX3 platform switches. For more information, refer to Configure PTP GM, on page 84 .

Step 5 Configure the optional commands as needed.

- Configure the domain number to use for this clock using the command [**no**] **ptp domain***number*

  PTP domains allow you to use multiple independent PTP clocking subdomains on a single network. The range for the *number* is from 0 to 127.

- Increase the number of PTP sessions by offloading some timers to the line card uisng the comand [**no**] **ptp offload**

  This step is required for one-step mode and optional for two-step mode.

  Note Make sure that neither of these features are already configured: dot1x (**feature dot1x**) and NV overlay (**feature nv overlay**). A dynamic ifacl label is reserved when these features are configured. However, only two dynamic ifacl label bits are available. If either of these features are already configured, a dynamic ifacl label will not be available for PTP offload, and the feature cannot be enabled. Note that PTP (**feature ptp**) consumes one ifacl label.

  Note Beginning with Cisco NX-OS Release 9.3(3), Cisco Nexus 9504 and 9508 platform switches with 9636C-R, 9636C-RX, or 9636Q-R line cards support offload only with one-step clock operation. PTP offload is automatically enabled or disabled when the one-step clock operation is enabled or disabled.

- Configure the PTP clock operation to the one-step mode using the command [**no**] **ptp clock-operation one-step**

  In this case, the timestamp message is sent as a part of the sync message. A followup message is not sent in this mode.

- Configure the priority1 value to use when advertising this clock using the command [**no**] **ptp priority1***value*

  This value overrides the default criteria (clock quality, clock class, and so on) for best master clock selection. Lower values take precedence. The range for the *value* is from 0 to 255.



Note For the switch to synchronize with an external Grand Master clock, the local switch PTP priority value must be configured higher than that of external Grand Master Clock priority.

- Configure the priority2 value to use when advertising this clock using the command [**no**] **ptp priority2***value*

  This value is used to decide between two devices that are otherwise equally matched in the default criteria. For example, you can use the priority2 value to give a specific switch priority over other identical switches. The range for the *value* is from 0 to 255.



Note For the switch to synchronize with an external Grand Master clock, the local switch PTP priority value must be configured higher than that of external Grand Master Clock priority.

Example:

```
switch(config)# ptp domain 1
switch(config)# ptp offload
switch(config)# ptp clock-operation one-step
switch(config)# ptp priority1 1
switch(config)# ptp priority2 1
```

Step 6 Configure support for PTP management packets using the command [**no**] **ptp management**

Example:

```
switch(config)# ptp management
switch(config-ptp-profile)#
```

This command is enabled by default. **no** : Disables support for management packets.

Step 7 Configure the optional commands as needed.

- Enable the PTP profile and enter the PTP profile configuration mode using the command [**no**] **ptp profile** { **8275-1** | **default** }



Note This command is supported beginning with Cisco NX-OS Release 9.3(5) and only on the N9K-C93180YC-FX3S platform switch. For more information, see Configure PTP Telecom Profile, on page 94 .

- Configure the PTP delay mean path/reverse path tolerance variation using the command [**no**] **ptp delay tolerance** { **mean-path** | **reverse-path** } *variation*

  mean-path : Ignore spikes in Mean Path Delay (MPD) as calculated by the PTP BMC algorithm.

  reverse-path : Ignore spikes in (t4-t3) as calculated by the PTP BMC algorithm.

  variation : Percentage that defines the tolerance for spikes. Use numeric values with a single decimal. Range is from 1.0 through 100.0.

  Note This command is supported beginning with Cisco NX-OS Release 9.3(5).

- Configure the switch to forward all PTPv1 packets based on the forwarding rule using the command **ptp forward-version1**

  Note If you do not enable this command, all PTPv1 packets are passed on to the CPU and ultimately dropped.

  This command is supported beginning with Cisco NX-OS Release 9.3(6).

- Enable PTP unicast sessions to transition to negotiated mode using the command **ptp unicast-negotiation**.

  This configuration is introduced in Cisco Nexus NX-OS Release 10.2(1)F and is supported on 93180YC-FX3S. From Cisco NX-OX Release 10.2(2)F onwards, this configuration is supported on Cisco Nexus 9300-FX, 9300-FX2, 9300-GX, and 9300-GX2 platform switches.

  For more information, refer to the PTP Unicast-Negotiation section.

- Set the transmit scope for PTP over IPv6 multicast configuration using the command **ptp ipv6 multicast scope tx**<*scope*>

  Note
  - Transmit scope is a single value. This determines the scope of IPv6 multicast destination address (FF0'X') used by PTP packets transmitted by the switch.
  - The default value is 1 (`0x1`).
  - When interface level configuration is applied, it takes precedence over the corresponding global configuration.

- Set the receive scope for PTP over IPv6 multicast configuration using the command **ptp ipv6 multicast scope rx**<*scope*>

Note

- Receive scope can be a single value or multiple values or a range of values. This determines the acceptable scope (FF0'X') for received IPv6 multicast PTP packets.
- The default value is 1 through 0xE (`0x1-0xE`).
- When interface level configuration is applied, it takes precedence over the corresponding global configuration.

- Enable the PTP time distribution hold feature using the command [**no**] **ptp time distribution-hold** [**correction-threshold** <corr_limit>] [**delay-threshold** <max_delay_time>]

  correction-threshold — Holds the time distribution until the correction settles down to the specified correction value in nanoseconds.

  delay-threshold — Sets the maximum time limit in seconds to hold the time-distribution. However, if correction threshold is met before the delay threshold, the time distribution resumes.

  Default correction threshold is 300 nanoseconds. Default delay threshold is 300 seconds for TOR and 900 seconds for modular chassis.

  The maximum correction threshold is 100000 nanoseconds. The maximum delay threshold is 5000 seconds.

- Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# ptp profile default
switch(config-ptp-profile)#

switch(config)# ptp delay tolerance mean-path 50.5
switch(config)#

switch(config)# ptp forward-version1
switch(config)#

switch(config)# ptp ipv6 multicast scope tx 0x2

switch(config)# ptp ipv6 multicast scope rx 0x1-0x5,0x8,0xA-0xE

switch(config)# ptp time distribution-hold correction-threshold 90000ns delay threshold 4000s

switch(config)# copy running-config startup-config
```

Configure PTP GM

Beginning with Cisco NX-OS Release 10.3(2)F, the following are the newly added CLIs to support PTP GM functionality:

Procedure

Step 1 Configure the device type as ordinary clock grandmaster using the command [**no**] **ptp device-typeordinary-clock-grandmaster**

Example:

```
switch(config)# ptp device-type ordinary-clock-grandmaster
```

Step 2 (Optional) Configure the **utc-offset** value for the PTP GM functionality using the command **ptp utc-offset** < *leap-seconds* > [**next-leapevent** < *date* > < *time* > **new-offset** < *new-leap-seconds* >]

Example:

```
switch(config)#
ptp utc-offset 37 next-leap-event 2022-04-30 23:59:59 new-offset 38
```

Default value is 37 seconds.

- **utc-offset** : UTC offset value to be added to GM clock.

  leap-seconds : Leap seconds value, range is 0–125 seconds. The default value is 37 seconds.

  next-leapevent : Optional keyword to configure time at which leap seconds change.

  date : Date at which leap seconds value changes in YYYY-MM-DD format.

  time : Time at which leap seconds value will change in HH:MM:SS format

  new-offset : New UTC offset value to be used after above configured **utc-offset** value.

  new-leap-seconds : New leap seconds value, range is 0–125 seconds. It can only be ± 1 of the configured **utc-offset** value.

Step 3 Enable the GNSS Time of Day (ToD) sync to system clock on the device using the command **clock protocol gnss**

Example:

```
switch(config)#
clock protocol gnss
```

Example:

Note For external GNSS configuration, refer to *Configuring GPS Settings for the Grandmaster Clock* section. For internal GNSS configuration, refer to Enable the GNSS receiver, on page 144 section.

Configure PTP on an Interface

After you globally enable PTP, it is not enabled on all supported interfaces by default. You must enable PTP interfaces individually.

Before you begin

Make sure that you have globally enabled PTP on the switch and configured the source IP address for PTP communication.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Specify the interface on which you are enabling the PTP and enter the interface configuration mode using the command **interface ethernet***slot/port*

Example:

```
switch(config)# interface ethernet 2/1
switch(config-if)#
```

Step 3 Enable or disable PTP on an interface using the command [**no**] **ptp**

Example:

```
switch(config-if)# ptp
```

Step 4 Configure the master clock that is assigned as the PTP role of the interface using the command **ptp multicast master-only**

Example:

```
switch(config)# ptp multicast master-only
switch(config)#
```

Note This command is deprecated in Cisco NX-OS Release 9.3(5) and is not supported in future releases.

Step 5 (Optional) Proceed with the following optionals commands as needed

- Display the PTP status using the command **show ptp brief**
- Display the status of the PTP port using the command **show ptp port interface***interface slot/port*
- Set the transmit scope for PTP over IPv6 multicast configuration using the command **ptp ipv6 multicast scope tx**<*scope*>

Note

- Transmit scope is a single value. This determines the scope of IPv6 multicast destination address (FF0'X') used by PTP packets transmitted by the switch.
- The default value is 1 (`0x1`).
- When interface level configuration is applied, it takes precedence over the corresponding global configuration.

- Set the receive scope for PTP over IPv6 multicast configuration using the command **ptp ipv6 multicast scope rx**<*scope*>

Note

- Receive scope can be a single value or multiple values or a range of values. This determines the acceptable scope (FF0'X') for received IPv6 multicast PTP packets.
- The default value is 1 through 0xE (`0x1-0xE`).
- When interface level configuration is applied, it takes precedence over the corresponding global configuration.
- Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config-if)# show ptp brief
switch(config-if)# show ptp port interface ethernet 2/1
switch(config)# ptp ipv6 multicast scope tx 0x2
switch(config)# ptp ipv6 multicast scope rx 0x1-0x5,0x8,0xA-0xE
switch(config-if)# copy running-config startup-config
```

Configure PTP in Unicast Mode

Configure Unicast Mode

Traditional PTP messages are delivered to the nodes that are capable of receiving PTP multicast messages. (For example, **announce**, **sync**, **delay_req**, **delay_resp** and **follow_up**). In Unicast mode, all PTP messages are delivered only to a particular PTP node. Multicast address is not used. In unicast mode, you can configure master/slave role and assign corresponding peer slave/master IP addresses.

Up to 8 master IPs can be configured for a slave unicast port and 64 slave IPs can be configured for a master port with a maximum 256 slave IP total for all ports. The following commands are used to configure the unicast slave IPs and unicast master IPs. Unicast packets are only sent to and received from these IPs. Packets received from other IPs are ignored.

For Cisco NX-OS Release 9.3(5) and later:

```
switch(config-if)# ptp
switch(config-if)# ptp transmission unicast
switch(config-if)# ptp role master
switch(config-if)# ptp slave 10.10.10.2

switch(config-if)# ptp
switch(config-if)# ptp transmission unicast
switch(config-if)# ptp role slave
switch(config-if)# ptp master 10.10.10.1
```

For Cisco NX-OS Release 9.3(4) and earlier:

```
switch(config-if)# ptp transport ipv4 ucast master
switch(config-if-ptp-master)# slave ipv4 10.10.10.2
```

```
switch(config-if)# ptp transport ipv4 ucast slave
switch(config-if-ptp-slave)# master ipv4 10.10.10.1
```

Configure Unicast Mode for IPv4 or IPv6

Traditional PTP messages are delivered to the nodes that are capable of receiving PTP multicast messages. (For example, **announce**, **sync**, **delay_req**, **delay_resp** and **follow_up**). In Unicast mode, all PTP messages are delivered only to a particular PTP node. Multicast address is not used. In unicast mode, you can configure master/slave role and assign corresponding peer slave/master IP addresses.

Up to 8 master IPs can be configured for a slave unicast port and 64 slave IPs can be configured for a master port with a maximum 256 slave IP total for all ports. The following commands are used to configure the unicast slave IPs and unicast master IPs. Unicast packets are only sent to and received from these IPs. Packets received from other IPs are ignored.

For Cisco NX-OS Release 10.2(1)F and later:

```
IPv4 config
interface Ethernet1/34
ptp
ptp transport ipv4
ptp transmission unicast
ptp role master
ptp slave 10.10.10.2
ptp ucast-source 10.10.10.1

interface Ethernet1/35
ptp
ptp transport ipv4
ptp transmission unicast
ptp role slave
ptp master 10.10.10.1
ptp ucast-source 10.10.10.2
```

```
IPv6 config
interface Ethernet1/34
ptp
ptp transport ipv6
ptp transmission unicast
ptp role master
ptp slave 2012:a1:0:0:0:0:0:2
ptp ucast-source 2012:a1:0:0:0:0:0:1

interface Ethernet1/35
ptp
ptp transport ipv6
ptp transmission unicast
ptp role slave
ptp master 2012:a1:0:0:0:0:0:1
ptp ucast-source 2012:a1:0:0:0:0:0:2
```

For Cisco NX-OS Release 9.3(5) and later:

```
switch(config-if)# ptp
switch(config-if)# ptp transmission unicast
switch(config-if)# ptp role master
switch(config-if)# ptp slave 10.10.10.2

switch(config-if)# ptp
switch(config-if)# ptp transmission unicast
switch(config-if)# ptp role slave
switch(config-if)# ptp master 10.10.10.1
```

For Cisco NX-OS Release 9.3(4) and earlier:

```
switch(config-if)# ptp transport ipv4 ucast master
switch(config-if-ptp-master)# slave ipv4 10.10.10.2

switch(config-if)# ptp transport ipv4 ucast slave
switch(config-if-ptp-slave)# master ipv4 10.10.10.1
```

Assign Dynamic Role

Complete the following steps to assign a dynamic role:

Procedure

Step 1 Enter the global configuration mode command **configure terminal**

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Specify the interface on which you enable PTP and enter the interface configuration mode using the following command **interface ethernet***slot/port*

Example:

```
switch(config)# interface ethernet 2/1
                        switch(config-if)#
```

Step 3 Enable or disable PTP on the interface using the command [**no**] **ptp**

Example:

```
switch(config-if)# ptp
                        switch(config-if)#
```

Note This command is required prior to applying below unicast configuration commands on the interface.

Step 4 Configure the PTP transmission method that is used by the interface using the command **ptp transmission unicast**

Example:

```
switch(config-if)# ptp transmission unicast
                        switch(config-if)#
```

Step 5 Configure the PTP role of the interface using the command **ptp role dynamic**

Example:

```
switch(config-if)# ptp role dynamic
                        switch(config-if)#
```

dynamic : The dynamic clock is assigned as the PTP role of the interface.

Step 6 Set the IP address of the peer clock when the PTP role of the interface is set to dynamic using the command**ptp peer***ipv4-address*

Example:

```
switch(config-if)# ptp peer 10.10.10.1
                        switch(config-if)#
```

Assign Master Role

Complete the following steps to assign a master role:

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Specify the interface on which you enable PTP and enter the interface configuration mode using the command**interface ethernet***slot/port*

Example:

```
switch(config)# interface ethernet 2/1
                        switch(config-if)#
```

Note After configuring this command, for Cisco NX-OS Release 9.3(5) and later, skip to step 5. For Cisco NX-OS Release 9.3(4) and earlier, continue with step 3.

Step 3 Enable PTP master on a particular port (Layer 3 interface) using the command [**no**] **ptp transport ipv4 ucast master**

Example:

```
switch(config-if)# ptp transport ipv4 ucast master
                        switch(config-if-ptp-master)#
```

In the master sub-mode, you can enter the slave IPv4 addresses.

Step 4 Enter the slave IPv4 addresses using the command **slave ipv4**<*IP_address*>

Example:

```
switch-1(config)# interface ethernet 1/1
                        switch-1(config-if)# ptp transport ipv4 ucast master
                        switch-1(config-if-ptp-master)# slave ipv4 1.2.3.1
                        switch-1(config-if-ptp-master)# slave ipv4 1.2.3.2
                        switch-1(config-if-ptp-master)# slave ipv4 1.2.3.3
                        switch-1(config-if-ptp-master)# slave ipv4 1.2.3.4
                        switch-1(config-if-ptp-master)#
```

Maximum of 64 IP addresses are allowed per master, but this number varies and it depends on the sync interval configuration. The master sends announce, sync, follow-up, and delay_resp only to these slave addresses. You must make sure that the slave IP is reachable.

Note For Cisco NX-OS Release 9.3(4) and earlier, this concludes the procedure.

Step 5 Enable or disable PTP on an interface using the command [**no**] **ptp**

Example:

```
switch(config-if)# ptp
                        switch(config-if)#
```

Note Starting with 9.3(5), this command is required prior to applying below unicast configuration commands on the interface.

Step 6 Configure the PTP transmission method that is used by the interface using the command **ptp transmission unicast**

Example:

```
switch(config-if)# ptp transmission unicast
                        switch(config-if)#
```

Note This command is supported beginning with Cisco NX-OS Release 9.3(5).

Step 7 Configure the PTP role of the interface using the command **ptp role master**

Example:

```
switch(config-if)# ptp role master
                        switch(config-if)#
```

master : The master clock is assigned as the PTP role of the interface.

Note This command is supported beginning with Cisco NX-OS Release 9.3(5).

Step 8 Set the IP address of the slave clock when the PTP role of the interface is set to "master" using the command **ptp slave***ipv4-address*

Example:

```
switch(config-if)# ptp slave 10.10.10.2
                        switch(config-if)#
```

Note This command is supported beginning with Cisco NX-OS Release 9.3(5).

Assign Slave Role

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Specify the interface on which you enable PTP and enter the interface configuration mode using the command**interface ethernet***slot/port*

Example:

```
switch(config)# interface ethernet 2/1
                        switch(config-if)#
```

Note After configuring this command, for Cisco NX-OS Release 9.3(5) and later, skip to step 5. For Cisco NX-OS Release 9.3(4) and earlier, continue with step 3.

Step 3 Enable PTP slave on a particular port (Layer 3 interface) using the command [**no**] **ptp transport ipv4 ucast slave**

Example:

```
switch(config-if)# ptp transport ipv4 ucast slave
                        switch(config-if-ptp-slave)#
```

In the slave sub-mode, you can enter the master IPv4 addresses.

Step 4 Enter the master IPv4 addresses using the command**master ipv4**<*IP_address*>

Example:

```
switch-1(config)# interface ethernet 1/1
switch-1(config-if)# ptp transport ipv4 ucast slave
switch-1(config-if-ptp-slave)# master ipv4 4.4.4.1
switch-1(config-if-ptp-slave)# master ipv4 4.4.4.2
switch-1(config-if-ptp-slave)# master ipv4 4.4.4.3
```

Note For Cisco NX-OS Release 9.3(4) and earlier, this concludes the procedure.

Step 5 Enable or disable PTP on an interface using the command [**no**] **ptp**

Example:

```
switch(config-if)# ptp
                      switch(config-if)#
```

Note Starting with 9.3(5), this command is required prior to applying below unicast configuration commands on the interface

Step 6 Configure the PTP transmission method that is used by the interface using the command **ptp transmission unicast**

Example:

```
switch(config-if)# ptp transmission unicast
                      switch(config-if)#
```

Note This command is supported beginning with Cisco NX-OS Release 9.3(5).

Step 7 Configure the PTP role of the interface using the command **ptp role slave**

Example:

```
switch(config-if)# ptp role slave
                      switch(config-if)#
```

slave : The slave clock is assigned as the PTP role of the interface.

Note This command is supported beginning with Cisco NX-OS Release 9.3(5).

Step 8 Set the IP address of the master clock when the PTP role of the interface is set to "slave" using the command **ptp master***ipv4-address*

Example:

```
switch(config-if)# ptp master 10.10.10.1
                      switch(config-if)#
```

Note This command is supported beginning with Cisco NX-OS Release 9.3(5).

Configure Unicast Source Address

Note For all releases up to, and including Cisco NX-OS Release 9.3(4), if the PTP configuration on the interface is changed from unicast to multicast or unicast slave to unicast master, you must reconfigure the unicast source address.

Beginning with Cisco NX-OS Release 9.3(5), if the PTP configuration on the interface is changed from unicast to multicast or unicast slave to unicast master, you do not need to reconfigure the unicast source address.

Complete the following steps to configure unicast source address:

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Specify the interface on which you enable the PTP and enter the interface configuration mode using the following command **interface ethernet***slot/port*

Example:

```
switch(config)# interface ethernet 2/1
                        switch(config-if)#
```

Step 3 Configure the PTP source address per interface level using the command[**no**] **ptp ucast-source** { <*ipv4-addr*> | <*ipv6-addr*> } [**vrf**<*vrf-name*>

Example:

```
switch(config-if)# ptp ucast-source 10.10.10.20
                        switch(config-if)#
```

This IP address is used only for unicast PTP messages. The PTP unicast source IP address must be reachable.

ipv4-address : The IPv4 address of the unicast source. This is used when transport is set to IPv4.

ipv6-address : The IPv6 address of the unicast source. This is used when transport is set to IPv6.

vrf*vrf-name* : If the IP address is in non-default VRF, use the vrf parameter to configure the VRF.

Configure PTP Telecom Profile

Configure Global PTP Telecom Profile

This procedure describes the steps involved to configure PTP telecom profile including the clock and its settings to be consistent with ITU-T Telecom Profiles for Frequency.

Before you begin

The QoS TCAM region Ingress SUP [ingress-sup] must be set to 768 or higher. Follow these steps:

1. Check the TCAM region by using the **show hardware access-list tcam region** command.
2. If the Ingress SUP region is not set to 768 or higher, then configure the Ingress SUP TCAM region using the command **hardware access-list tcam region ing-sup 768** . Copy the running configuration to the startup configuration (**copy running-config startup-config**) and reload the switch.

Procedure

Step 1 Enters global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Enable the global PTP feature using the command **feature ptp**

Example:

```
switch(config)# feature ptp
                        switch(config)#
```

Step 3 Configure the PTP operational mode for the switch using the command Profile Default: **mode** { **hybrid** | **non-hybrid** | **none** }

Example:

```
switch(config)# mode hybrid
                        switch(config-ptp-profile)#
```

The two operational modes for the switch are:

hybrid : The SyncE source acts as the PTP source.

default : The local/1588 clock acts as the PTP source.

Note

- This command is automatically configured when the **ptp profile** command is set. The configuration value cannot be changed. See #unique_104 unique_104_Connect_42_step_gtc_cts_tkb for more information.
- The only mode that the 8275-2 profile supports is non-hybrid .
- The profile default mode is hybrid.

Step 4 Exit the PTP profile configuration mode and returns to the global configuration mode using the command **exit**

Example:

```
switch(config-ptp-profile)# exit
                        switch(config)#
```

Step 5 Configure the source IPv4 address for all the PTP packets in the multicast PTP mode using the command **ptp source***ip-address*

Example:

```
switch(config)# ptp source 10.10.10.20
                        switch(config)#
```

Step 6 Configure the priority value to use when advertising this clock using the command Profile Default: **ptp***priority value**value*

Example:

```
switch(config)# ptp priority1 128
                        switch(config)#

switch(config)# ptp priority2 128
                        switch(config)#
```

The priority value can be priority1 or priority2. This value overrides the default criteria (clock quality, clock class, etc.) for best master clock selection. Lower values take precedence. The default value is 128 and the range is from 0 to 255.

Note This command is automatically configured when the **ptp profile 8275-1 or 8275-2** global command is set. The configuration value cannot be changed. See #unique_104 unique_104_Connect_42_step_gtc_cts_tkb .

Step 7 Configure the peer delay request interval using the command **ptp pdelay-req-interval***value*

Example:

```
switch(config)# ptp pdelay-req-interval 0
                        switch(config)#
```

value : Range is from 0 through 5.

Step 8 Specify the PTP clock domain value using the command Profile Default: **ptp domain***value*

Example:

```
switch(config)# ptp domain 24
                        switch(config)#
```

The allowed domain number range for G.8275.1 profile is between 24 and 43. Default is 24. The allowed domain number range for G.8275.2 profile is between 44 and 63. Default is 44.

Note This command is automatically configured when the **ptp profile 8275-1** or **8275-2** global command is set. See #unique_104 unique_104_Connect_42_step_gtc_cts_tkb .

Configure PTP Telecom Profile 8275-1 on an Interface

This procedure describes the steps that are involved to configure PTP telecom profile 8275-1 for interfaces.

Note Some commands that are described in this procedure are automatically enabled and configured when the **ptp profile 8275-1** global command is set and PTP is enabled on the interface. See Configure Global PTP Telecom Profile, on page 94 for more information.

Before you begin

This procedure along with configuring frequency synchronization on the interface, constitutes the required interface settings for the "hybrid PTP" platform. For more information about the interface frequency synchronization configuration, see Configure Frequency Synchronization on an interface, on page 55 .

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Specify the interface on which you configure PTP telecom profile parameters and enter the interface configuration mode using the command **interface ethernet***slot* / *port*. Enable PTP on the interface using the command [**no**] **ptp**

Example:

```
switch(config)# interface ethernet 1/5
                        switch(config-if)#

switch(config-if)# ptp
                        switch(config-if)#
```

Step 3 Specify the transport mechanism that is used to send PTP packets using the command : Profile Default: **ptp transportethernet** and configure the PTP transmission method that is used by the interface using the command : Profile Default: **ptp transmission multicast**

Example:

```
switch(config-if)# ptp transport ethernet
                        switch(config-if)#

switch(config-if)# ptp transmission multicast
                        switch(config-if)#
```

For **ethernet** , PTP packets are carried only in Eth frame (Eth/ptp). For **multicast** , PTP uses multicast destination IP address 224.0.1.129 as per IEEE 1588 standards for communication between devices.

Note This command is automatically configured when the **ptp profile 8275-1** global command is set. For more information about the **ptp profile 8275-1** command, see Configure Global PTP Telecom Profile, on page 94.

Step 4 Configure the PTP role of the interface using the command : Profile Default: **ptp role dynamic**

Example:

```
switch(config-if)# ptp role dynamic
                        switch(config-if)#
```

For **dynamic** , the best master clock algorithm (BMCA) assigns the role.

Note This command is automatically configured when the **ptp profile 8275-1** global command is set. For more information about the **ptp profile 8275-1** command, see Configure Global PTP Telecom Profile, on page 94.

Step 5 Accept and respond to nonforwardable destination MAC address packets using the command **ptp destination-mac non-forwardable rx-no-match accept**

Example:

```
switch(config-if)# ptp destination-mac non-forwardable rx-no-match accept
                        switch(config-if)#
```

This is an optional step. These destination MAC addresses are used in the PTP messages that are exchanged between the GM clock, PTP-master clock, and PTP-slave clocks.

Step 6 Configure the value used in the BMCA's selection of the best master clock using the command Profile Default: **ptp cost***value* and configures the minimum interval that is allowed between PTP delay messages when the port is in the master state using the command Profile Default: **ptp delay-request minimum interval***log-seconds*

Example:

```
switch(config-if)# ptp cost 128
                        switch(config-if)#


switch(config-if)# ptp delay-request minimum interval -4
```

If all the parameters mentioned in the standard are the same, then this local priority is used.

Note This command is automatically configured when the **ptp profile 8275-1** global command is set. For more information about the **ptp profile 8275-1** command, see Configure Global PTP Telecom Profile, on page 94.

Step 7 Configure the interval between PTP announce messages on an interface or the number of PTP intervals before a timeout occurs on an interface using the command Profile Default: **ptp announce interval***log-seconds* and configures the interval between PTP synchronization messages on an interface using the command Profile Default: **ptp sync interval***log-seconds*

Example:

```
switch(config-if)# ptp announce interval -3
switch(config-if)# ptp sync interval -4
```

Note This command is automatically configured when the **ptp profile 8275-1** global command is set. For more information about the **ptp profile 8275-1** command, see Configure Global PTP Telecom Profile, on page 94.

Step 8 Configure the number of PTP intervals before a timeout occurs on an interface using the command [**no**] **ptp announcetimeout***count*

Example:

```
switch(config-if)# ptp announce timeout 3
```

This is an optional step. The range for the PTP announcement interval timeout is from 2 to 4 intervals.

Step 9 To change the following commands in this interface configuration using the command [**no**] **ptp profile-override**

Example:

```
switch(config-if)# ptp profile-override
                       switch(config-if)#
```

This is an optional step that is disabled by default. When enabled, it allows you to change the following commands in this interface configuration:

- **ptp transport**
- **ptp announce interval**
- **ptp delay-request minimum interval**
- **ptp sync interval**
- **ptp cost** (8275-1 profile only)

Note When enabled, changes to the commands will not be reset to default if the global PTP profile is changed. Removing **ptp profile-override** resets the PTP configuration on the interface to the default values corresponding to the global profile.

Configure PTP Telecom Profile 8275-2 on an Interface

This procedure describes the steps that are involved to configure PTP telecom profile 8275-2 for interfaces.

Note Some commands that are described in this procedure are automatically enabled and configured when the **ptp profile 8275-2** global command is set and PTP is enabled on the interface. See Configure Global PTP Telecom Profile, on page 94 for more information.

Before you begin

This procedure along with configuring frequency synchronization on the interface, constitutes the required interface settings for the "hybrid PTP" platform. For more information about the interface frequency synchronization configuration, see Configure Frequency Synchronization on an interface, on page 55 .

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal** and specify the interface on which you are configuring PTP telecom profile parameters and enter the interface configuration mode using the command **interface ethernet***slot* / *port*

Example:

```
switch# configure terminal
                        switch(config)#

switch(config)# interface ethernet 1/5
                        switch(config-if)#
```

Step 2 Enable the PTP on the interface using the command [**no**] **ptp** and specify the transport mechanism that is used to send PTP packets using the profile default command **ptp transportipv4**

Example:

```
switch(config-if)# ptp
                        switch(config-if)#

switch(config-if)# ptp transport ipv4
                        switch(config-if)#
```



Note This command is automatically configured when the **ptp profile 8275-2** global command is set. For more information about the **ptp profile 8275-2** command, see Configure Global PTP Telecom Profile, on page 94

Step 3 Configure the PTP transmission method that is used by default by the interface using the command **ptp transmission unicast** and configure the default PTP role of the interface using the command **ptp role dynamic**

Example:

```
switch(config-if)# ptp transmission unicast
                        switch(config-if)#

switch(config-if)# ptp role dynamic
                        switch(config-if)#
```

For **unicast**, PTP uses the configured master/slave/peer IP. For **dynamic** , the best master clock algorithm (BMCA) assigns the role.

Note This command is automatically configured when the **ptp profile 8275-2** global command is set. For more information about the **ptp profile 8275-2** command, see Configure Global PTP Telecom Profile, on page 94.

Step 4 Accept and respond to non-forwardable destination MAC address packets using the command **ptp destination-mac non-forwardable rx-no-match accept**

Example:

```
switch(config-if)# ptp destination-mac non-forwardable rx-no-match accept
                      switch(config-if)#
```

These destination MAC addresses are used in the PTP messages that are exchanged between the GM clock, PTP-master clock, and PTP-slave clocks.

Step 5 Configure the value used in the BMCA's selection of the best master clock using the command **ptp cost***value* and configure the minimum interval that is allowed between PTP delay messages when the port is in the master state using the command **ptp delay-request minimum interval***log-seconds*

Example:

```
switch(config-if)# ptp cost 128
                      switch(config-if)#

switch(config-if)# ptp delay-request minimum interval -4
```

If all the parameters mentioned in the standard are the same, then this local priority is used.

Note This command is automatically configured when the **ptp profile 8275-2** global command is set. For more information about the **ptp profile 8275-2** command, see Configure Global PTP Telecom Profile, on page 94.

Step 6 Configure the interval between PTP announce messages on an interface or the number of PTP intervals before a timeout occurs on an interface using the command **ptp announce interval***log-seconds*

Example:

```
switch(config-if)# ptp announce interval -3
```

Note This command is automatically configured when the **ptp profile 8275-2** global command is set. For more information about the **ptp profile 8275-2** command, see Configure Global PTP Telecom Profile, on page 94.

Step 7 Configure the interval between PTP synchronization messages on an interface using the profile default command : **ptp sync interval***log-seconds*

Example:

```
switch(config-if)# ptp sync interval -4
```

Note This command is automatically configured when the **ptp profile 8275-2** global command is set. For more information about the **ptp profile 8275-2** command, see Configure Global PTP Telecom Profile, on page 94 .

Step 8 Configure the number of PTP intervals before a timeout occurs on an interface using the command [**no**] **ptp announcetimeout***count*

Example:

```
switch(config-if)# ptp announce timeout 3
```

This is an optional step and the range for the PTP announcement interval timeout is from 2 to 4 intervals.

Step 9 To change any of the commands mentioned below you can use this command : [**no**] **ptp profile-override**

Example:

```
switch(config-if)# ptp profile-override
                        switch(config-if)#
```

This is an optional step that is disabled by default. You can enable it to change the following commands in this interface configuration:

- **ptp transport**
- **ptp announce interval**
- **ptp delay-request minimum interval**
- **ptp sync interval**
- **ptp cost** (8275-2 profile only)

Note When enabled, changes to the commands will not be reset to default if the global PTP profile is changed. Removing **ptp profile-override** resets the PTP configuration on the interface to the default values corresponding to the global profile.

Verify PTP Telecom Profile Configuration

After performing the PTP telecom profile configuration tasks, use this reference to verify the configuration.

show running-config ptp all

The output of this command displays global and interface configurations for PTP telecom profile.

The following is an example of the output of the **show running-config ptp all** command:

```
switch# show running-config ptp all
!Command: show running-config ptp all
!Running configuration last done at: Fri Feb 21 20:09:55 2020
!Time: Fri Feb 21 21:10:19 2020
```

```
version 9.3(5) Bios:version 01.00
feature ptp

ptp profile 8275-1
  mode hybrid
ptp source 0.0.0.0
ptp device-type boundary-clock
ptp priority1 128
ptp priority2 10
ptp pdelay-req-interval 0
no ptp notification type parent-change
no ptp notification type gm-change
no ptp notification type high-correction
no ptp notification type port-state-change
ptp correction-range 100000
no ptp correction-range logging
ptp management
ptp mean-path-delay 1000000000
ptp domain 24
ttag-marker-interval 60


interface Ethernet1/1
  ptp
  no ptp profile-override
  ptp destination-mac non-forwardable rx-no-match accept
  ptp transport ethernet
  ptp transmission multicast
  ptp role dynamic
  ptp cost 128
  ptp delay-request minimum interval -4
  ptp announce interval -3
  ptp sync interval -4
  ptp announce timeout 3

interface Ethernet1/6
  ptp
  no ptp profile-override
  ptp destination-mac non-forwardable rx-no-match accept
  ptp transport ethernet
  ptp transmission multicast
  ptp role dynamic
  ptp cost 128
  ptp delay-request minimum interval -4
  ptp announce interval -3
  ptp sync interval -4
  ptp announce timeout 3

interface Ethernet1/7
  ptp
  no ptp profile-override
  ptp destination-mac non-forwardable rx-no-match accept
  ptp transport ethernet
  ptp transmission multicast
  ptp role dynamic
  ptp cost 128
  ptp delay-request minimum interval -4
  ptp announce interval -3
  ptp sync interval -4
  ptp announce timeout 3

interface Ethernet1/8
  ptp
```

```
no ptp profile-override
ptp destination-mac non-forwardable rx-no-match accept
ptp transport ethernet
ptp transmission multicast
ptp role dynamic
ptp cost 128
ptp delay-request minimum interval -4
ptp announce interval -3
ptp sync interval -4
ptp announce timeout 3
```

Note The output of the **show running-config ptp all** command displays a complete list of all the PTP configured interfaces.

show ptp parent

The output of this command displays the properties of a PTP parent.

The following is an example of the output of the **show ptp parent** command:

```
switch# show ptp parent
PTP PARENT PROPERTIES

Parent Clock:
Parent Clock Identity: 10:b3:d6:ff:fe:bf:a8:63
Parent Port Number: 0
Observed Parent Offset (log variance): N/A
Observed Parent Clock Phase Change Rate: N/A

Grandmaster Clock:
Grandmaster Clock Identity: 10:b3:d6:ff:fe:bf:a8:63
Grandmaster Clock Quality:
        Class: 248
        Accuracy: 254
        Offset (log variance): 65535
        Priority1: 128
        Priority2: 10
```

show ptp corrections

The output of this command displays up to the last 2000 correction details for each PTP slave port.

The following is an example of the output of the **show ptp corrections** command:

```
switch# show ptp corrections
PTP past corrections
--------------------------------------------------------------------------------
Slave Port            SUP Time                Correction(ns)    MeanPath Delay(ns)
----------  ------------------------------  ------------------  ------------------
Eth1/3      Thu Feb 20 22:51:02 2020 861523           4           260
Eth1/3      Thu Feb 20 22:51:02 2020 735961           4           260
Eth1/3      Thu Feb 20 22:51:02 2020 610170           4           268
Eth1/3      Thu Feb 20 22:51:02 2020 483106           0           280
Eth1/3      Thu Feb 20 22:51:02 2020 355745           0           280
Eth1/3      Thu Feb 20 22:51:02 2020 229924          -4           268
Eth1/3      Thu Feb 20 22:51:02 2020 104819          -4           268
Eth1/3      Thu Feb 20 22:51:01 2020 979604           8           272
```

show ptp clock

The output of this command displays the properties of the local clock, including clock identity.

The following is an example of the output of the **show ptp clock** command:

```
switch# show ptp clock
PTP Device Type : boundary-clock
PTP Device Encapsulation : NA
PTP Source IP Address : 0.0.0.0
Clock Identity : 10:b3:d6:ff:fe:bf:a8:63
Clock Domain: 24
Slave Clock Operation : Unknown
Master Clock Operation : Two-step
Slave-Only Clock Mode : Disabled
Number of PTP ports: 35
Priority1 : 128
Priority2 : 10
Clock Quality:
        Class : 248
        Accuracy : 254
        Offset (log variance) : 65535
Offset From Master : 0
Mean Path Delay : 0
Steps removed : 0
Correction range : 100000
MPD range : 1000000000
Local clock time : Wed Feb 26 17:08:34 2020
Hardware frequency correction : NA
PTP Clock state               : Free-Run
```

show ptp brief

The output of this command displays the PTP clock state for each configured port.

The following is an example of the output of the **show ptp brief** command:

```
switch# show ptp brief
PTP port status
-----------------------------------
Port                  State
--------------------- ------------
Eth1/1                Slave
Eth1/6                Disabled
Eth1/7                Disabled
Eth1/8                Disabled
Eth1/10               Master
Eth1/11               Disabled
Eth1/12               Disabled
Eth1/13               Master
Eth1/14               Disabled
Eth1/15               Disabled
Eth1/16               Disabled
Eth1/17               Disabled
Eth1/18               Disabled
Eth1/19               Disabled
Eth1/20               Disabled
Eth1/21               Disabled
Eth1/22               Disabled
Eth1/23               Disabled
Eth1/24               Disabled
Eth1/25               Disabled
Eth1/26               Disabled
Eth1/27               Disabled
```

```
Eth1/28               Disabled
Eth1/29               Disabled
Eth1/30               Disabled
Eth1/31               Disabled
Eth1/32               Disabled
Eth1/33               Disabled
Eth1/34               Disabled
Eth1/35               Disabled
Eth1/36               Disabled
Eth1/37               Disabled
Eth1/38               Disabled
Eth1/39               Disabled
Eth1/40               Disabled
```

show ptp clock foreign-masters record

The output of this command displays the state of foreign masters known to the PTP process. For each foreign master, the output displays the clock identity, basic clock properties, and whether the clock is being used as a grandmaster..

The following is an example of the output of the **show ptp clock foreign-master-record** command:

```
switch# show ptp port status
P1=Priority1, P2=Priority2, C=Class, A=Accuracy,
OSLV=Offset-Scaled-Log-Variance, SR=Steps-Removed
GM=Is grandmaster

---------   ----------------------  ---  ----  ----  ---  -----  --------
Interface         Clock-ID          P1   P2    C     A    OSLV   SR
---------   ----------------------  ---  ----  ----  ---  -----  --------

Eth1/1    00:00:00:00:00:00:00:01    128  128   6     33   65535  0    GM
```

show ptp detail

The output of this command displays the list of all connected peers for each PTP port and indicates whether the role is static or dynamic.

```
switch(config-if)# show ptp detail
PTP port details:
    Flags:   (S) Static, (D) Dynamic
    Connected Peer details: IP  clock-id

Port             State         Transmission     Transport        Flags
====             =====         ============     =========        =====
Eth1/27/1        Slave         Unicast          IPV4             (S)
Connected Peers:
    10.10.1.36           00:fd:22:ff:fe:8d:1a:a5-0x1901

Eth1/27/2        Passive       Unicast          IPV4             (D)
Connected Peers:
    10.10.2.36           00:00:00:00:00:00:00:00-0x0

Eth1/27/3        Master        Multicast        IPV4             (D)
Connected Peers:
```

Configure PTP Notifications

You can enable, disable, and customize notifications for the following significant PTP events:

- Change in the Grand Master (GM) clock

- Change in the Parent clock
- Change in the PTP state on a port
- High PTP clock corrections

The notifications are generated by the DME infrastructure based on information it receives from PTP.

Procedure

Step 1 Configure the system to send a change notification if any PTP clock changes using the command [**no**] **ptp notification type** *clock type*-**change**

Example:

```
switch(config)# ptp notification type <clock type>-change
                        switch(config)#
```

This step can be used to a notification for a grand master or a parent clock.

Step 2 Configure the system to send a notification if a port state change event occurs using the command [**no**] **ptp notification type port-state-change** [**category** { **all** | **master-slave-only** }] [**interval** { **immediate** | *seconds* [**periodic-notification** { **disable** | **enable** }] }]

Example:

```
switch(config)# ptp notification type port-state-change category master-slave-only
                        switch(config)#
```

- **category** : Specifies which state changes must occur for a notification to be sent.
  - **all** : Every port state change is reported.

    Note Using the **all** option results in many notifications.

  - **master-slave-only** : Port state changes from and to the master-slave state are only reported.
- **interval***seconds* : Port state change notifications are sent at the configured interval: from 1-300 seconds with a granularity of 1 sec.
  - **periodic-notification** : Determines if periodic notifications are sent even if a port state change has not occurred during the configured interval.

    disable : A port state change notification is reported only if the current state is not the same as the previously reported state. Any intermediate state changes during the configured periodic interval are ignored. For example, if a port is a MASTER at time X, and changes to DISABLED and then back to MASTER by the time X+periodic-interval occurs, then no notification is generated for the intervening events.

    enable : Port state change notifications are sent at the configured interval, irrespective of a change in the port state.
- **interval immediate** : A port State Change Notification is sent when the state changes.

Step 3 Configure the system to send a high-correction notification if a PTP high correction event occurs using the command [**no**] **ptp notification type high-correction** [**interval** { *seconds* [**periodic-notification** { **disable** | **enable** }] | **immediate** }]

Example:

```
switch(config)# ptp notification type high-correction interval immediate
                        switch(config)#
```

A high correction event is when the correction exceeds the value that is configured in the **ptp correction-range** command (see the following optional step).

- **interval***seconds* : High-correction notifications are sent at the configured interval: 1–300 seconds with a granularity of 1 second.
  - **periodic-notification** : Determines if periodic notifications are sent even if any high correction has not occurred during the configured interval.

    disable : Send a notification only if high correction events occurred during the configured periodic interval. This is the default setting.

    enable : Send a notifications irrespective of the number of high correction events during the configured periodic interval. If there are no such events, the payload indicates zero high correction events during the periodic interval.
- **interval immediate** : Send a notification as soon as a high correction event occurs.

The high correction notification contains the following attributes:

- highCorrectionCount
- lastHighCorrectionTime
- lastHighCorrectionValue
- lastHighCorrectionMPD
- maxHighCorrectionTime
- maxHighCorrectionValue
- maxHighCorrectionMPD

Step 4 Configure a threshold that, once exceeded, indicates that a PTP high correction has occurred using the command [**no**] **ptp correction-range** { *nanoseconds* | **logging** }

Example:

```
switch(config)# ptp correction-range 200000
                        switch(config)#
```

This is an optional step. Range is 10–1000000000. The default is 100000 (100 microseconds).

Configure PTP Interface to Stay in Master State

This procedure describes how to prevent an endpoint from causing a port to transition to a slave state.

Before you begin

- Make sure that you have globally enabled PTP on the switch and configured the source IP address for PTP communication.
- After you globally enable PTP, it is not enabled on all supported interfaces by default. You must enable PTP interfaces individually.

Procedure

Step 1 Enter global configuration mode using the command : **switch # configure terminal**

Step 2 Specify the interface on which you are enabling PTP and enters interface configuration mode using the command **switch(config) # interface ethernet***slot/port*

Step 3 Enable or disable PTP on an interface using the command **switch(config) # ptp**



Note After configuring this command, for Cisco NX-OS Release 9.3(5) and later, skip to step 5. For Cisco NX-OS Release 9.3(4) and earlier, continue with step 4.

Step 4 Configure the port to maintain the master state using the command **switch(config-if) #***ptpmulticastmaster-only* **ptpmulticastmaster-only**

Example:

```
switch(config)#
show ptp brief
PTP port status
-----------------------------------
Port                  State
----------------  ----------------
Eth1/1              Slave
switch(config)#
interface ethernet 1/1
switch(config-if)#
ptp multicast master-only

2001 Jan  7 07:50:03 A3-MTC-CR-1 %$ VDC-1 %$ %PTP-2-PTP_GM_CHANGE: Grandmaster clock has changed
                from 60:73:5c:ff:fe:62:a1:41 to 58:97:bd:ff:fe:0d:54:01 for the PTP protocol

2001 Jan  7 07:50:03 A3-MTC-CR-1 %$ VDC-1 %$ %PTP-2-PTP_STATE_CHANGE: Interface Eth1/1 change from
                PTP_BMC_STATE_SLAVE to PTP_BMC_STATE_PRE_MASTER

2001 Jan  7 07:50:03 A3-MTC-CR-1 %$ VDC-1 %$ %PTP-2-PTP_TIMESYNC_LOST: Lost sync with  master clock

2001 Jan  7 07:50:07 A3-MTC-CR-1 %$ VDC-1 %$ %PTP-2-PTP_STATE_CHANGE: Interface Eth1/1 change from
                PTP_BMC_STATE_PRE_MASTER to PTP_BMC_STATE_MASTER
```

Note This command is supported in Cisco NX-OS Release 9.3(4) and earlier. It is deprecated in Cisco NX-OS Release 9.3(5) and later.

For Cisco NX-OS Release 9.3(4) and earlier, this concludes the procedure.

Step 5 Configure the port to maintain the master state using the command **ptp role master**

Note This command is supported beginning with Cisco NX-OS Release 9.3(5).

Verify PTP Configuration

Use one of the following commands to verify the configuration:

Table 6: PTP Show Commands

Command	Purpose
show ptp brief	Displays the PTP status.
show ptp clock	Displays the properties of the local clock, including clock identity.
show ptp clock foreign-masters-record	Displays the state of foreign masters known to the PTP process. For each foreign master, the output displays the clock identity, basic clock properties, and whether the clock is being used as a grandmaster.
show ptp corrections	Displays the last few PTP corrections.
show ptp counters [**all** | **interface ethernet** *slot/port*]	Displays the PTP packet counters for all interfaces or for a specified interface.
show ptp parent	Displays the properties of the PTP parent.
show ptp port interface ethernet *slot/port*	Displays the status of the PTP port on the switch.
show ptp time-property	Displays the PTP clock properties.
show running-config ptp [**all**]	Displays the running configuration for PTP.
clear ptp counters [**all** | **interface ethernet** *slot/port*]	Clears all PTP messages that are received and transmitted on a specific interface or on all interfaces that has PTP enabled.

Command	Purpose
show ptp detail	Displays the list of all connected peers for each PTP port and indicates whether the role is static or dynamic.

Configuration Examples

This example shows how to configure PTP globally on the device, specify the source IP address for PTP communications, and configure a preference level for the clock:

```
switch# configure terminal
switch(config)# feature ptp
switch(config)# ptp source 10.10.10.1
switch(config)# ptp priority1 1
switch(config)# ptp priority2 1
switch(config)# show ptp brief
PTP port status
-----------------------
Port State
------- --------------
switch(config)# show ptp clock
PTP Device Type: Boundary clock
Clock Identity : 0:22:55:ff:ff:79:a4:c1
Clock Domain: 0
Number of PTP ports: 0
Priority1 : 1
Priority2 : 1
Clock Quality:
  Class : 248
  Accuracy : 254
  Offset (log variance) : 65535
Offset From Master : 0
Mean Path Delay : 0
Steps removed : 0
Local clock time:Mon Dec 22 14:13:24 2014
```

This example shows how to configure PTP on an interface and configure the intervals for the announce, delay-request, and synchronization messages:

```
switch# configure terminal
switch(config)# interface Ethernet 1/1
switch(config-if)# ptp
switch(config-if)# ptp announce interval 3
switch(config-if)# ptp announce timeout 2
switch(config-if)# ptp delay-request minimum interval smpte-2059-2 -3
switch(config-if)# ptp sync interval smpte-2059-2 -3
switch(config-if)# no shutdown
switch(config-if)# show ptp brief
PTP port status
-----------------------
Port State
------- --------------
Eth2/1 Master
switch(config-if)# show ptp port interface ethernet 2/1
PTP Port Dataset: Eth2/1
Port identity: clock identity: 0:22:55:ff:ff:79:a4:c1
Port identity: port number: 1028
```

```
PTP version: 2
Port state: Master
Delay request interval(log mean): 4
Announce receipt time out: 2
Peer mean path delay: 0
Announce interval(log mean): 3
Sync interval(log mean): 1
Delay Mechanism: End to End
Peer delay request interval(log mean): 0
```

This example shows how to configure master/slave role and assign corresponding peer slave/master IP addresses.

```
For Cisco NX-OS Release 9.3(5) and later:
switch# configure terminal
switch(config)# interface ethernet 1/1
switch(config-if)# ptp
switch(config-if)# ptp transmission unicast
switch(config-if)# ptp role master
switch(config-if)# ptp slave 10.1.1.2
switch(config-if)# ptp ucast-source 11.0.0.1
switch(config-if)# ip address 11.0.0.1/24
switch(config-if)# no shutdown

switch# configure terminal
switch(config)# interface ethernet 1/1
switch(config-if)# ptp
switch(config-if)# ptp transmission unicast
switch(config-if)# ptp role slave
switch(config-if)# ptp master 10.1.1.2
switch(config-if)# ptp ucast-source 11.0.0.1
switch(config-if)# ip address 11.0.0.1/24
switch(config-if)# no shutdown

For Cisco NX-OS Release 9.3(4) and earlier:
switch-1(config)# interface ethernet 1/1
switch-1(config-if)# ptp transport ipv4 ucast master
switch-1(config-if-ptp-master)# slave ipv4 1.2.3.1
switch-1(config-if-ptp-master)# slave ipv4 1.2.3.2
switch-1(config-if-ptp-master)# slave ipv4 1.2.3.3
switch-1(config-if-ptp-master)# slave ipv4 1.2.3.4
switch-1(config-if-ptp-master)#

switch-1(config-if)# ptp transport ipv4 ucast slave
switch-1(config-if-ptp-slave)# master ipv4 4.4.4.1
switch-1(config-if-ptp-slave)# master ipv4 4.4.4.2
switch-1(config-if-ptp-slave)# master ipv4 4.4.4.3


switch-1(config-if-ptp-slave)# ptp ucast-source 9.9.9.9


switch-1(config-if)# sh running-config ptp

!Command: show running-config ptp
!Time: Tue Feb  7 17:37:09 2017

version 7.0(3)I4(6)
feature ptp

ptp source 1.1.1.1
```

```
interface Ethernet1/1
  ptp transport ipv4 ucast master
    slave ipv4 1.2.3.1
    slave ipv4 1.2.3.2
    slave ipv4 1.2.3.3
    slave ipv4 1.2.3.4

interface Ethernet1/2
  ptp transport ipv4 ucast slave
    master ipv4 4.4.4.1
    master ipv4 4.4.4.2
    master ipv4 4.4.4.3
  ptp ucast-source 9.9.9.9

switch-1(config-if)#
```

This example shows how to configure PTP in clock operation mode with master or slave ports.

```
PLTFM-A(config)# show ptp clock
PTP Device Type : boundary-clock
PTP Device Encapsulation : layer-3
PTP Source IP Address : 1.1.1.1
Clock Identity : 74:26:ac:ff:fe:fd:de:ff
Clock Domain: 0
Slave Clock Operation : One-step
Master Clock Operation : One-step
Slave-Only Clock Mode : Disabled
Number of PTP ports: 142
Priority1 : 200
Priority2 : 200
Clock Quality:
        Class : 248
        Accuracy : 254
        Offset (log variance) : 65535
Offset From Master : -32
Mean Path Delay : 105
Steps removed : 1
Correction range : 200
MPD range : 100
Local clock time : Wed Jul  3 18:57:23 2019
Hardware frequency correction : NA
```

PTP Unicast-Negotiation

Enabling PTP unicast transmission is a pre-requisite for using unicast-negotiation.

Beginning with Cisco NX-OS Release 10.2(1)F, PTP Telecom Profile is supported only on N9K-C93180YC-FX3S platform.

Beginning with Cisco NX-OS 10.2(1)F release, the following are the newly added CLIs:

```
ptp unicast-negotiation
     schema <name>
            [ no ] announce interval <log-val>
            [ no ] sync interval <log-val>
            [ no ] delay-response interval <log-val>
            [ no ] announce duration  <seconds>  [renew-offset <seconds>]
            [ no ] sync duration  <seconds> [renew-offset <seconds>]
            [ no ] delay-response duration  <seconds> [renew-offset <seconds>]
```

```
            [ no ] announce interval range <minimum-log-val> <maximum-log-val>
            [ no ] sync interval range <minimum-log-val> <maximum-log-val>
            [ no ] delay-response interval range <minimum-log-val> <maximum-log-val>
            [ no ] announce duration range <minimum-seconds> <maximum-seconds>
            [ no ] sync duration range <minimum-seconds> <maximum-seconds>
            [ no ] delay-response duration range <minimum-seconds> <maximum-seconds>

interface ethernet 1/34
     [ no ] ptp master {<ipv4-addr> | <ipv6-addr>} [negotiation-schema <schema-name>]
     [ no ] ptp transmission unicast negotiation-schema <WORD>

show ptp unicast-negotation
            show ptp unicast-negotiation interface eth <port>
            show system internal ptp unicast-negotation
```

Enable PTP Unicast-Negotiation

Enabling PTP unicast transmission is a pre-requisite for using unicast-negotiation.

Beginning with Cisco NX-OS 10.2(1)F release, the following are the newly added CLIs:

Procedure

Step 1 Create a default schema using the command **switch (config-ptp-ucast-negotiation)# schema** <*schema-name*>

A default schema will be created when unicast-negotiation is enabled and will be applied to all interfaces that have PTP-unicast enabled and to the master IPs currently configured. Schema name can be any string up to 31 characters.

Step 2 Configure the interval between PTP announce messages using the command **switch (config-ptp-ucast-nego-schema)# announce interval** <*log-seconds*>

The range varies from -3 to 0. Default value is -1.

Step 3 Configure the interval between PTP synchronization messages using the command **switch (config-ptp-ucast-nego-schema)# sync interval** <*log-seconds*>

This is an optional step. The range varies from -4 to 0. Default value is -3.

Step 4 Configure the interval allowed between PTP delay messages when the port is in master state using the command **switch (config-ptp-ucast-nego-schema)# delay-response interval** <*log-seconds*>

This is an optional step. The range varies from -4 to 0. Default value is -2.

Step 5 Configure the duration of any type of session using the command **switch (config-ptp-ucast-nego-schema)#***type of session***duration** <*seconds*> [**renew-offset** <*seconds*>]

This is an optional step which can be used to configure an announce, sync or delay-response session.

To set how soon the slave sends renewal request for the session use the command **renew-offset** <*seconds*> :

Default value is 10 which means, it will send renewal request 10 seconds before expiry of session (granted duration). The range is 60 to 1000. Default value is 300.

Step 6 Configure an acceptable range of values for any type of interval requests from slave using the command **switch (config-ptp-ucast-nego-schema)#** *type of interval request***interval range** <*minimum-log-val*> <*maximum-log-val*>

This is an optional step which can be used to configure acceptable range of values for announce, sync or delay-response interval requests. Default for minimum-log-val is -3. Default for maximum-log-val is 0.

Step 7 Configure an acceptable range of values for any type of session duration requests from slave using the command**switch (config-ptp-ucast-nego-schema)#** *type of session duration request* **duration range**<*minimum-seconds*> <*maximum-seconds*>

This is an optional step which can be used to configure acceptable range of values for announce, sync or delay-response session duration requests. Default for minimum-seconds is 60. Default for maximum-seconds is 1000.

Step 8 To show the status of unicast-negotiation use the command, **show ptp unicast-negotiation** [*interface ethernet slot/port*]

Enhanced Multicast Scale

This feature is to be used only in specific deployment scenarios where higher scaling of PTP multicast secondary devices is required even though the ability to debug is very limited.

This feature has the following limitations:

- The high scale in the number of PTP slaves implies very high PTP control packet rate. As a result, copp rate needs to be increased appropriately. For more information about Configuring Control Plane Policing, refer to the appropriate version of the *Cisco Nexus 9000 Series NX-OS Security Configuration Guide* on cisco.com.
- PTP debugs should be completely disabled using the **no ptp debug all** command, along with various internal logs, for example, PTP internal syslogs. As a result, ability to debug issues will be very limited.
- We recommend that PTP secondary port should not share hardware MAC (port fifo) with any of the scaled PTP primary ports. Additionally, not more than 2 primary ports should be enabled per hardware MAC. The hardware MAC for ports on any given switch can be checked using the following command:

  show interface hardware-mappings
- In rare occasions, the corrections can spike to the milliseconds range.

Perform the following command to enable the scaling of PTP multicast secondary devices:

ptp enhanced-client-scale

To view the status of the above command, run the following command:

switch# **show run ptp** | **grep enhanced**

Timestamp Tagging

The timestamp tagging feature provides precision time information to track in real time when packets arrive at remote devices. Packets are truncated and timestamped using PTP with nanosecond accuracy. Using the TAP aggregation functionality on the switch, along with the Cisco Nexus Data Broker, you can copy the network traffic using SPAN, filter and timestamp the traffic, and send it for recording and analysis.

If you configure **ttag** on an interface, all incoming traffic will be tagged. If you configure **ttag-strip** on an interface all outgoing traffic with ttag will be removed.

Configure Timestamp Tagging

Note Configuring timestamp tagging is not supported on Cisco Nexus 9508 switches with 9636C-R, 9636C-RX, and 9636Q-R line cards.

Note

- When you use the ttag feature in a VXLAN EVPN multisite deployment, make sure that the ttag is stripped (**ttag-strip**) on BGW's DCI interfaces that connect to the cloud. To elaborate, if the ttag is attached to non-Nexus 9000 devices that do not support ether-type 0x8905, stripping of ttag is required. If no stripping is done, the non-Nexus devices will drop the packet.
- BGW back-to-back model of DCI does not require ttag stripping.
- For Cloudscale platforms, if a packet is received with a TTAG header (ethertype 0x8905), it will be preserved, unless **ttag-strip** is configured on the outgoing interface. After switching or routing, the packet egressing the switch will also have a TTAG header. For this to happen, the switch does not need to have any PTP or TTAG-related configuration. If the next hop would be a non-Cloudscale platform, such as a Catalyst switch - it might not recognize ether-type 0x8905, and could drop the packet.
- Cisco Nexus 9800 switches do not support routing of ether-type 0x8905 packets.

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Enter the interface configuration mode for the specified interface using the command **interface***type slot/port*

Example:

```
switch(config)# interface ethernet 2/2
switch(config-if)#
```

Step 3 Configure the timestamp tagging on the Layer 2 or Layer 3 egress interface using the command [**no**] **ttag**

Example:

```
switch(config-if)# ttag
```

This is required on the ingress port for the traffic that needs to be tagged when egressing the switch. This is not required on the egress port.

Configure TTAG Marker Packets and Time Interval

The ttag timestamp field attaches a 48-bit timestamp on the marker packet. This 48-bit timestamp is not a human familiar ASCII based timestamp. To make this 48-bit timestamp human readable, the ttag marker packet can be used to provide additional information to decode the 48-bit timestamp information.

Field	Position (byte:bit)	Length	Definition
Magic		16	By default, this field displays A6A6. This enables to identify ttag-marker packets on the packet stream.
Version		8	Version number. The default version is 1.
Granularity		16	This field represents the granularity of the 48-bit timestamp size. By default, the value is 04, which is 100 picoseconds or 0.1. nanoseconds.
UTc_offset		8	The utc_offset between the ASIC and the UTC clocks. The default value is 0.
Timestamp_hi		32	The high 16-bit of 48- bit ASIC hardware timestamp. **Note** Add Correction_hi and Correction_lo to Timestamp_hi and Timestamp_lo fields to get the 64-bit ASIC hardware timestamp.
Timestamp_lo		32	The low 32-bit of 48- bit ASIC hardware timestamp. **Note** Add Correction_hi and Correction_lo to Timestamp_hi and Timestamp_lo fields to get the 64-bit ASIC hardware timestamp.
UTC sec		32	The seconds part of UTC timestamp from the CPU clock of the Cisco Nexus 9000 Series switch.
UTC nsec		32	The nanoseconds part of UTC timestamp from the CPU clock of the Cisco Nexus 9000 Series switch.
Reserved		32	Reserved for future use.

Correction_hi		32	The high 32-bit of cumulative PTP correction on the Cisco Nexus 9000 Series switch. **Note** Add Correction_hi and Correction_lo to Timestamp_hi and Timestamp_lo fields to get the 64-bit ASIC hardware timestamp.
Correction_lo		32	The low 32-bit of cumulative PTP correction on the Cisco Nexus 9000 Series switch. **Note** Add Correction_hi and Correction_lo to Timestamp_hi and Timestamp_lo fields to get the 64-bit ASIC hardware timestamp.
Signature		32	The default value is 0xA5A5A5A5. This allows a forward search of marker packet and provide references to the UTC timestamp, so the client software can use that reference UTC to recover the 32-bit hardware timestamp in each packet header.
Pad		8 64	This is align byte to convert the ttag-marker align to 4 byte boundary.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Configure the seconds that a switch will take to send a ttag-marker packet to the outgoing ports using the command **ttag-marker-interval***seconds*

Example:

```
switch(config-if)# ttag-marker-interval 90
```

This is a global setting to the switch. By default, it sends a ttag-marker packet every 60 seconds. The range for seconds is from 1 to 25200.

Step 3 Enter the interface configuration mode for the specified interface using the command **interface***type slot/port*

Example:

```
switch(config)# interface ethernet 2/2
switch(config-if)#
```

Step 4 Send the ttag-marker packets to the outgoing port using the command [**no**] **ttag-marker enable**

Example:

```
switch(config-if)# ttag-marker enable
```

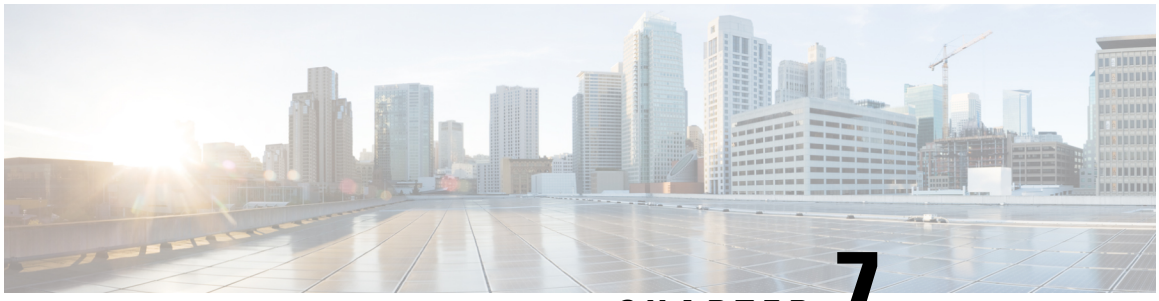
Step 5 Remove TTAG from egress packets on the interface using the command **ttag-strip**

Example:

```
switch(config-if)# ttag-strip
```

Related Documents

Related Topic	Document Title
1588 IEEE	1588 IEEE standards



CHAPTER 7

Configuring GPS

This chapter describes the GPS input and how to configure it.

This chapter includes the following sections:

- Global Positioning System, on page 121
- Guidelines and limitations for GPS, on page 121
- Configure GPS settings, on page 122
- Commands for verifying GPS configuration, on page 123

Global Positioning System

The router can receive 1PPS, 10 MHz, and ToD signals from an external clocking and timing source.

The router can receive 1PPS, 10 MHz, and ToD signals from an external clocking and timing source. The three inputs are combined as a Sync-2 interface to form the external timing source or the Global Positioning System (GPS) input.

The GPS front panel connector details are:

- ToD—RS422 format as input
- 1PPS—RS422 or DIN connector as input
- 10MHz—DIN connector as input

GPS input starts only when all the three signals – 1PPS, 10MHz, and ToD are UP.

Note Unlike the Ethernet interface, the Sync-2 interface cannot receive or transmit QL. Ensure that you assign a QL value to the Sync-2 interface.

By default, 1PPS and 10MHz are in output mode. ToD output mode is not configurable.

Guidelines and limitations for GPS

The guidelines and limitations for GPS are:

- Beginning with NX-OS Release 10.3(2)F, the GPS input is supported only on the Nexus 93180YC-FX3 switch with these limitations:
  - The Time of Day (ToD) output is not supported.
  - RS422 ToD input is not supported.
    - GPS without RS422 TOD input can be used only for frequency and phase synchronization.

Configure GPS settings

Use this procedure to configure GPS settings for the grandmaster clock and enable GPS as input.

Procedure

Step 1 Enter global configuration mode using the **configure terminal** command.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Enter the GPS clock configuration mode using the **clock-interface sync 1/1** command.

Example:

```
switch(config)# clock-interface sync 1/1
switch(config-clk-if)# port-parameters
switch(config-clk-params)#
```

Step 3 Enable the GPS as input and configure the source for PPS input and format for the Time of Day (ToD) input using the **gps-input tod-format** *value* **pps-input** *value* command.

Example:

```
switch(config-clk-params)# gps-input tod-format ntp4 pps-input ttl
switch(config-clk-params)# exit
switch(config-clk-if)#
```

Supported ToD formats are:

Incoming ToD Input	Format
`cisco`	ASCII format
`ntp4`	NTP Type4 format

Supported PPS input formats are:

Incoming PPS Input	Format
RS422	RS422 format
TTL	TTL format

Step 4 Configure the frequency synchronization on GPS using the **frequency synchronization** command.

Example:

```
switch(config-clk-if)# frequency synchronization
switch(config-clk-freqsync)# selection input
switch(config-clk-freqsync)# wait-to-restore 0
switch(config-clk-freqsync)# exit
switch(config-clock-if)#
```

Commands for verifying GPS configuration

After performing the GPS configuration tasks, use this reference to check for configuration errors and to verify the configuration.

show clock-interface

The output of this command displays details of the GPS clock configuration.

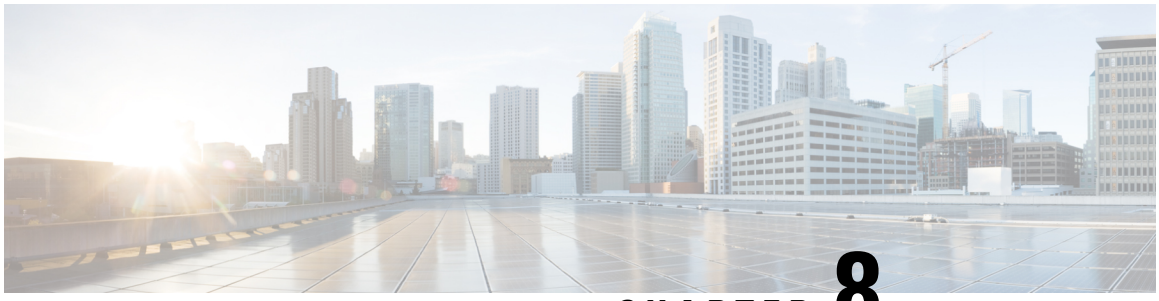
This is an example output of the **show frequency synchronization** selection command.

```
switch(config-clk-if)# sh frequency synchronization selection
==============
Selection point: System Clock (T0) Selector (2 inputs, 1 selected)
Last programmed 00:53:56 ago, and selection made 00:53:35 ago
Next selection points
Node scoped :
Uses frequency selection
Used for local line interface output
S Input Last Selection Point QL Pri Status
== ======================= ======================= ===== === ===========
Internal0[1] n/a SEC 255 Available
11 Sync1[1] n/a PRC 100 Locked
==============
Selection point: IEEE 1588 Clock Selector (2 inputs, 1 selected)
Last programmed 00:53:56 ago, and selection made 00:53:55 ago
Next selection points
Node scoped :
Uses frequency selection
S Input Last Selection Point QL Pri Status
== ======================= ======================= ===== === ===========
21 Internal0[1] n/a SEC 255 Holdover
Sync1[1] n/a PRC 100 Unmonitored
==============
switch(config-clk-if)#
```

When the front panel timing LED is Green, it indicates that the GPS is configured and 1PPS, ToD, and 10M inputs are valid.

This table shows the default LED status for GPS Timing based on the input to the device.

TIMING	Off	GPS config and GPS port is down. Time-of-day (ToD), 1PPS, and 10-MHz ports are not provisioned or disabled.
	Amber	ToD, 1PPS, and 10-MHz signals are not valid.
	Green	GPS port is up. ToD, 1PPS, and 10-MHz signals are valid.



CHAPTER 8

Network Time Protocol

The Network Time Protocol (NTP) synchronizes the time of day among a set of distributed time servers and clients so that you can correlate events when you receive system logs and other time-specific events from multiple network devices. NTP uses the User Datagram Protocol (UDP) as its transport protocol. All NTP communications use Coordinated Universal Time (UTC).

- Overview, on page 125
- Guidelines and Limitations, on page 127
- Enable or Disable NTP, on page 130
- Configure Device as an Authoritative NTP Server, on page 133
- Configure Server and Peer, on page 133
- Configure Authentication, on page 135
- Configure Access Restrictions, on page 137
- Configure Source IP Address, on page 139
- Configure Source Interface, on page 139
- Configure Logging, on page 140
- Additional References, on page 141

Overview

A Network Time Protocol (NTP) server is a device that receives its time from an authoritative time source and distributes this time across the network to synchronize devices.

- It receives time from authoritative sources such as radio clocks, atomic clocks, or GPS time sources.
- It distributes synchronized time to network devices with high efficiency (as little as one packet per minute for millisecond accuracy).
- It uses a stratum system to describe the distance between a device and the authoritative time source.

Stratum and Synchronization Behavior

NTP uses a stratum system to indicate how far a device is from an authoritative time source, and compares time from multiple devices before synchronizing.

- A stratum 1 time server is directly attached to an authoritative time source (such as a radio or atomic clock or a GPS time source).

- A stratum 2 NTP server receives its time through NTP from a stratum 1 time server.

Before synchronizing, NTP compares the time reported by several network devices and does not synchronize with one that is significantly different, even if it is a stratum 1 server.

- If the network is isolated from the Internet, Cisco NX-OS allows you to configure the time as though it were synchronized through NTP, even though it was not.

The time kept on a device is a critical resource, so it is strongly recommended to use the security features of NTP to avoid accidental or malicious setting of incorrect time. Two mechanisms are available: an access list-based restriction scheme and an encrypted authentication mechanism.



Note You can create NTP peer relationships to designate the time-serving hosts that you want your network device to consider synchronizing with and to keep accurate time if a server failure occurs.

Associations

An NTP association can be one of the following:

- A peer association—The device can either synchronize to another device or allow another device to synchronize to it.
- A server association—The device synchronizes to a server.

You need to configure only one end of an association. The other device can automatically establish the association.

NTP as Time Server

The Cisco NX-OS device can use NTP to distribute time. Other devices can configure it as a time server. You can also configure the device to act as an authoritative NTP server, enabling it to distribute time even when it is not synchronized to an outside time source.

Clock Manager

Clocks are resources that need to be shared across different processes. Multiple time synchronization protocols, such as NTP, might be running in the system.

The clock manager allows you to specify the protocol to control the various clocks in the system. Once you specify the protocol, the system clock starts updating. For information on configuring the clock manager, see the Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide.

Virtualization Support

NTP recognizes virtual routing and forwarding (VRF) instances. NTP uses the default VRF if you do not configure a specific VRF for the NTP server and NTP peer. See the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide for more information about VRFs.

High Availability

Stateless restarts are supported for NTP. After a reboot or a supervisor switchover, the running configuration is applied. For more information on high availability, see the Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide.

You can configure NTP peers to provide redundancy in case an NTP server fails.

Default Settings

The following table lists the default settings for NTP parameters.

Parameters	Default
NTP	Enabled
NTP authentication	Disabled
NTP access	Enabled
NTP logging	Disabled

Guidelines and Limitations

To configure NTP, you must have connectivity to at least one server that is running NTP.

Configuration Guidelines

1. Configure **clock protocol ntp** to synchronize the system clock from an NTP server, or to act as an NTP server providing time to other devices.
2. We recommend that you configure a peer association with another device only when you are sure that your clock is reliable (which means that you are a client of a reliable NTP server).
3. A peer that is configured alone takes on the role of a server and should be used as a backup. If you have two servers, you can configure several devices to point to one server and the remaining devices to point to the other server. You can then configure a peer association between these two servers to create a more reliable NTP configuration.
4. If you have only one server, we recommend that you configure all the devices as clients to that server.
5. You can configure up to 64 NTP entities (servers and peers).
6. If you configure NTP in a VRF, ensure that the NTP server and peers can reach each other through the configured VRFs.

General Guidelines

- NTP server functionality is supported.
- Manually distribute NTP authentication keys on the NTP server and Cisco NX-OS devices across the network.
- If you are using the switch as an edge device and want to use NTP, we recommend using the **ntp access-group** command and filtering NTP only to the required edge devices.

- If the system has been configured with the **ntp passive** , **ntp broadcast client** , or **ntp multicast client** commands, when NTP receives an incoming symmetric active, broadcast, or multicast packet, it can set up an ephemeral peer association in order to synchronize with the sender.

Note Make sure that you specify **ntp authenticate** before enabling any of the preceding commands. Failure to do so will allow your device to synchronize with any device that sends one of the preceding packet types, including malicious attacker-controlled devices.

- If you specify the **ntp authenticate** command, when a symmetric active, broadcast, or multicast packet is received, the system does not synchronize to the peer unless the packet carries one of the authentication keys that are specified in the **ntp trusted-key** global configuration command.
- To prevent synchronization with unauthorized network hosts, the **ntp authenticate** command should be specified any time the **ntp passive** , **ntp broadcast client** , or **ntp multicast client** command has been specified unless other measures, such as the **ntp access-group** command, have been taken to prevent unauthorized hosts from communicating with the NTP service on the device.
- While configuring NTP authentication password, use printable ASCII characters without #. If the password contains # authentication will not work for that key.

  To configure NTP authentication passwords with length greater than 20 characters, use hex characters (0-9,A-F) only.
- The **ntp authenticate** command does not authenticate peer associations that are configured via the **ntp server** and **ntp peer** configuration commands. To authenticate the **ntp server** and **ntp peer** associations, specify the **key** keyword.
- A maximum of four IP ACLs can be configured for a single NTP access group. IPv4 and IPv6 ACLs are supported.
- If packet flooding occurs on the inband ports, it can increase the CPU usage by NTPD to more than 90%. To overcome this high CPU usage by NTPD, use the custom CoPP policy to rate limit the incoming traffic to NTP. For more information about creating a custom CoPP policy, refer to the Configuring Control Plane Policing chapter in the relevant version of the Cisco Nexus 9000 Series NX-OS Security Configuration Guide on cisco.com .

Note The recommended rate limit is 1000 kbps for the policy **CIR** field and 64,000 bytes for the **BC** field.

Release Specific Guidelines

Beginning with Cisco NX-OS Release 9.3(9)

- NTP version 4 (NTPv4) is supported on Nexus standalone switches.

Beginning with Cisco NX-OS Release 10.1(1)

- Cisco Nexus 9000 switches do not sync with stratum 14 and 15.

- NTP version 4 (NTPv4) is supported on Nexus standalone switches.

Beginning with Cisco NX-OS Release 10.3(3)F

- In line with the RFC 8573 standards, NTP security is enhanced with the AES128CMAC authentication mechanism along with Type-6 encryption support for authentication keys.
- This feature provides an option to set the password as Type-0, Type-7, or Type-6.
- The maximum number of unique keys which can be configured is 1024, and the range is 1 to 65535.
- For Type-6 authentication to work, configure the same primary (master) key on the device, which was used to generate the new Type-6 key being configured, along with **feature password encryption aes** .
- If you enforce re-encryption using the **encryption re-encrypt obfuscated** command, all the NTP non-Type-6 passwords will be re-encrypted to Type-6.

  The **encryption delete type6** command deletes all the Type-6 passwords configured on NTP.

  The **encryption decrypt type6** command decrypts the existing configured Type-6 passwords.
- To perform ISSD from AES128CMAC/Type-6 supported version to a non-AES128CMAC/Type-6 supported version, unconfigure the Type-6 keys and then perform the ISSD.
- It is recommended to specify the encryptType and keyString when programmatically (restconf/Netconf and so on) configuring a key chain. If not specified, then the key chain infra uses the already available (or default) value of the missing property to configure the keyString.
- If you need to configure with a property missing, then you need to follow the same sequence of steps in both the peer routers.

Beginning with Cisco NX-OS Release 10.4(4)M

- Before configuring name-based NTP servers (FQDN), a user must configure relevant name servers under the required VRFs. While configuring any FQDN-based NTP server or peer, the configuration is accepted whether the IPs are resolvable or not. If any FQDNs remain unresolved, the system tries to resolve them in the background at regular intervals until all such FQDNs are resolved or the configuration is removed. FQDN resolution can take up to a few minutes.

  A configured FQDN server must be reachable and must return the valid IP for the NTP server functionality to work smoothly. Use the **show ntp peers** or the **show ntp peer-status** command to verify the resolved IPs of FQDNs.

Best Practices for NTP configuration

Follow these best practices when configuring NTP:

- Configure multiple redundant NTP time sources to avoid reliance on a single server and improve reliability.
- Use a loopback interface as the NTP source for consistency across device reboots and interface changes.
- Enable NTP authentication to secure communications and prevent malicious time alterations.
- Monitor NTP status and logs regularly; implement detection and alerting mechanisms using commands like **show ntp status** and **show ntp associations**.

- Use the **ntp panic-threshold-apply** command to check the configuration sanity checks, which reject large clock changes and prevent network instability.
- Address failure modes related to time-based authentication keychains by configuring overlapping key lifetimes to avoid routing disruptions.
- Regularly monitor and validate NTP synchronization status with monitoring tools and SNMP traps.
- Ensure operational practices include multiple NTP sources and configuration checks to prevent routing disruptions caused by time inaccuracies.

These practices help maintain accurate time synchronization, improve NTP security, and minimize the risk of network instability caused by incorrect time settings.

Enable or Disable NTP

NTP is enabled by default.



Note Be aware that the Cisco NX-OS commands for this feature may differ from those commands used in Cisco IOS.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Enable or disable NTP using the command [**no**] **feature ntp**

Example:

```
switch(config)# feature ntp
```

Step 3 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

This example for md5 shows how to configure the device to synchronize only to time sources that provide authentication key 42 in their NTP packets:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# ntp authentication-key 42 md5 aNiceKey
switch(config)# ntp server 192.0.2.105 key 42
switch(config)# ntp trusted-key 42
switch(config)# ntp authenticate
switch(config)# copy running-config startup-config
[########################################] 100%
switch(config)#
```

This example for aes128cmac shows how to configure the device to synchronize only to time sources that provide authentication key 12 in their NTP packets:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# ntp authentication-key 12 aes128cmac password 0/6/7
switch(config)# ntp server 192.0.2.105 key 12
switch(config)# ntp trusted-key 12
switch(config)# ntp authenticate
switch(config)# copy running-config startup-config
[########################################] 100%
switch(config)#
```

This example shows an NTP access group configuration with the following restrictions:

- Peer restrictions are applied to IP addresses that pass the criteria of the access list named "peer-acl."
- Serve restrictions are applied to IP addresses that pass the criteria of the access list named "serve-acl."
- Serve-only restrictions are applied to IP addresses that pass the criteria of the access list named "serve-only-acl."
- Query-only restrictions are applied to IP addresses that pass the criteria of the access list named "query-only-acl."

```
switch# configure terminal
switch(config)# ntp peer 10.1.1.1
switch(config)# ntp peer 10.2.2.2
switch(config)# ntp peer 10.3.3.3
switch(config)# ntp peer 10.4.4.4
switch(config)# ntp peer 10.5.5.5
switch(config)# ntp peer 10.6.6.6
switch(config)# ntp peer 10.7.7.7
switch(config)# ntp peer 10.8.8.8
switch(config)# ntp access-group peer peer-acl
switch(config)# ntp access-group serve serve-acl
switch(config)# ntp access-group serve-only serve-only-acl
switch(config)# ntp access-group query-only query-only-acl
switch(config)# ip access-list peer-acl
switch(config-acl)# 10 permit ip host 10.1.1.1 any
switch(config-acl)# 20 permit ip host 10.8.8.8 any
switch(config)# ip access-list serve-acl
switch(config-acl)# 10 permit ip host 10.4.4.4 any
switch(config-acl)# 20 permit ip host 10.5.5.5 any
switch(config)# ip access-list serve-only-acl
switch(config-acl)# 10 permit ip host 10.6.6.6 any
switch(config-acl)# 20 permit ip host 10.7.7.7 any
switch(config)# ip access-list query-only-acl
```

```
switch(config-acl)# 10 permit ip host 10.2.2.2 any
switch(config-acl)# 20 permit ip host 10.3.3.3 any
```

Note When only a single ACL group is applied, then all the packets relevant for other ACL categories are denied and only packets relevant for the configured ACL group is processed, as mentioned in below scenarios:

- If serve ACL is configured, then only client, control, and private packets are processed and all the other packets are denied.
- If serve-only ACL is configured, then only client packets are processed and all the other packets are denied.

If more than a single ACL is configured, it follows the order of processing as mentioned in below scenario:

- If serve and serve-only both are configured for the same IP address without match-all configured, where the IP is permitted in serve-acl and denied in serve-only, the client, control, private packets are permitted for that IP.

What to do next

To display the NTP configuration, perform one of the following tasks:

Command	Purpose
show ntp access-groups	Displays the NTP access group configuration.
show ntp authentication-keys	Displays the configured NTP authentication keys.
show ntp authentication-status	Displays the status of NTP authentication.
show ntp logging-status	Displays the NTP logging status.
show ntp peer-status	Displays the status for all NTP servers and peers.
show ntp peers	Displays all the NTP peers.
show ntp rts-update	Displays the RTS update status.
show ntp source	Displays the configured NTP source IP address.
show ntp source-interface	Displays the configured NTP source interface.
show ntp statistics {**io** | **local** | **memory** | **peer** {**ipaddr** {*ipv4-addr* | *ipv6-addr*} | **name** *peer-name*}}	Displays the NTP statistics.
show ntp trusted-keys	Displays the configured NTP trusted keys.
show running-config ntp	Displays NTP information.

Use the **clear ntp session** command to clear the NTP sessions.

Use the **clear ntp statistics** command to clear the NTP statistics.

Configure Device as an Authoritative NTP Server

You can configure the device to act as an authoritative NTP server, enabling it to distribute time even when it is not synchronized to an existing time server.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure the device as an authoritative NTP server using the command [**no**] **ntp master** [*stratum*]

Example:

```
switch(config)# ntp master
```

You can specify a different stratum level from which NTP clients get their time synchronized. The range is from 1 to 15.

Step 3 Display the NTP configuration using the command **show running-config ntp**

Example:

```
switch(config)# show running-config ntp
```

This is an optional step.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure Server and Peer

Configure an NTP server and peer to synchronize the device clock with a reliable time source.

You can configure an NTP server and peer.

Before you begin

Make sure you know the IP address or Domain Name System (DNS) names of your NTP server and its peers.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Form an association with a server using the command [**no**] **ntp server** { *ip-address* | *ipv6-address* | *dns-name* } [**key** *key-id*] [**maxpoll** *max-poll*] [**minpoll** *min-poll*] [**prefer**] [**use-vrf** *vrf-name*]

Example:

```
switch(config)# ntp server 192.0.2.10
```

Use the **key** keyword to configure a key to be used while communicating with the NTP server. The range for the *key-id* argument is from 1 to 65535.

Use the **maxpoll** and **minpoll** keywords to configure the maximum and minimum intervals in which to poll a server. The range for the *max-poll* and *min-poll* arguments is from 4 to 16 (configured as powers of 2, so effectively 16 to 65536 seconds), and the default values are 6 and 4, respectively (*maxpoll* default = 64 seconds, *minpoll* default = 16 seconds).

Use the **prefer** keyword to make this server the preferred NTP server for the device.

Use the **use-vrf** keyword to configure the NTP server to communicate over the specified VRF. The *vrf-name* argument can be **default** , **management** , or any case-sensitive, alphanumeric string up to 32 characters.



Note If you configure a key to be used while communicating with the NTP server, make sure that the key exists as a trusted key on the device.

Step 3 Form an association with a peer using the command [**no**] **ntp peer** { *ip-address* | *ipv6-address* | *dns-name* } [**key** *key-id*] [**maxpoll** *max-poll*] [**minpoll** *min-poll*] [**prefer**] [**use-vrf** *vrf-name*]

Example:

```
switch(config)# ntp peer 2001:0db8::4101
```

You can specify multiple peer associations.

Use the **key** keyword to configure a key to be used while communicating with the NTP peer. The range for the *key-id* argument is from 1 to 65535.

Use the **maxpoll** and **minpoll** keywords to configure the maximum and minimum intervals in which to poll a server. The range for the *max-poll* and *min-poll* arguments is from 4 to 17 (configured as powers of 2, so effectively 16 to 131072 seconds), and the default values are 6 and 4, respectively (*maxpoll* default = 64 seconds, *minpoll* default = 16 seconds).

Use the **prefer** keyword to make this peer the preferred NTP peer for the device.

Use the **use-vrf** keyword to configure the NTP peer to communicate over the specified VRF. The *vrf-name* argument can be **default** , **management** , or any case-sensitive, alphanumeric string up to 32 characters.

Step 4 Display the configured server and peers using the command **show ntp peers**

Example:

```
switch(config)# show ntp peers
```

This is an optional step.



Note A domain name is resolved only when you have a DNS server configured.

When DNS/Name Server resolves both IPv4 and IPv6, IPv6 Address is preferred by NX-OS.

Step 5 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure Authentication

Use this task to configure NTP authentication on your device, ensuring that the device synchronizes only with trusted time sources.

You can configure the device to authenticate the time sources to which the local clock is synchronized. When you enable NTP authentication, the device synchronizes to a time source only if the source carries one of the authentication keys specified by the **ntp trusted-key** command. The device drops any packets that fail the authentication check and prevents them from updating the local clock. NTP authentication is disabled by default.

Before you begin

Make sure that you configured the NTP server with the authentication keys that you plan to specify in this procedure.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Define the authentication keys using the command [**no**] **ntp authentication-key** *number* { **md5** | **aes128cmac** } *password string encryption-type*

Example:

```
switch(config)# ntp authentication-key 42 md5 aNiceKey
```

```
switch(config)# ntp authentication-key 21 md5
 JDYk3pp/Fuv0zWyVSRhS6EDERSSsp1uA7s57dvdsx
 g74ndf02lEI9dF6WX6Z78/5R8qPmSRRrDUDtCcUlZ
 XDUrf0ErodS3ikPQA= 6
```

```
switch(config)# ntp authentication-key
 12 aes128cmac
 JDYkzj4NojJdSkQPvBhFvAO9xCSVwj2iRGvShNSg
 ER4JwMBMtUEibfqkscgZ4+/iTdDmeCRW9SGWLxKb
 3Xk5g8pz4bR7Iu1a7QA= 6
```

You can choose either md5 or aes128cmac authentication scheme.

If the user is using the Type-6 key generated from the same primary (**master**) key, then the device does not synchronize to a time source until the user enables **feature password encryption aes** .

For Type-0 and Type-7 encryption types, the maximum length is 32 characters. Until Release 10.3(3)F, it was 15 (alphanumeric) characters. For the Type-6 encryption type, the maximum limit is 128 characters.

Step 3 Form an association with a server using the command **ntp server** *ip-address* **key** *key-id*

Example:

```
switch(config)# ntp server 192.0.2.1 key 1001
```

Use the **key** keyword to configure a key to be used while communicating with the NTP server. The range for the *key-id* argument is from 1 to 65535.

To require authentication, the **key** keyword must be used. Any **ntp server** or **ntp peer** commands that do not specify the **key** keyword will continue to operate without authentication.

Step 4 Display the configured NTP authentication keys using the command **show ntp authentication-keys**

Example:

```
switch(config)# show ntp authentication-keys
```

This is an optional step.

Step 5 Specify one or more keys using the command [**no**] **ntp trusted-key** *number*

Example:

```
switch(config)# ntp trusted-key 42
```

This command specifies one or more keys (defined in Step 2) that an unconfigured remote symmetric, broadcast, and multicast time source must provide in its NTP packets in order for the device to synchronize to it. The range for trusted keys is from 1 to 65535.

This command provides protection against accidentally synchronizing the device to a time source that is not trusted.

Step 6 Display the configured NTP trusted keys using the command **show ntp trusted-keys**

Example:

```
switch(config)# show ntp trusted-keys
```

This is an optional step.

Step 7 Enable or disable authentication for ntp passive, ntp broadcast client, and ntp multicast using the command [**no**] **ntp authenticate**

Example:

```
switch(config)# ntp authenticate
```

NTP authentication is disabled by default.

Step 8 Display the status of NTP authentication using the command **show ntp authentication-status**

Example:

```
switch(config)# show ntp authentication-status
```

This is an optional step.

Step 9 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure Access Restrictions

You can control access to NTP services by using access groups. Specifically, you can specify the types of requests that the device allows and the servers from which it accepts responses.

If you do not configure any access groups, NTP access is granted to all devices. If you configure any access groups, NTP access is granted only to the remote device whose source IP address passes the access list criteria.

- Without the **match-all** keyword, the packet gets evaluated against the access groups (in the order mentioned below) until it finds a permit. If a permit is not found, the packet is dropped.
- With **match-all** keyword, the packet gets evaluated against all the access groups (in the order mentioned below) and the action is taken based on the last successful evaluation (the last access group where an ACL is configured).

- peer—process client, symmetric active, symmetric passive, serve, control, and private packets (all types)
- serve—process client, control, and private packets
- serve-only—process client packets only
- query-only—process control and private packets only

The access groups are evaluated in the following order:

1. peer (all packet types)
2. serve (client, control, and private packets)
3. serve-only (client packets) or query-only (control and private packets)

ACL processing of serve-only or query-only depends on the NTP packet type.

Procedure

Step 1 Enter global configuration mode uisng the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Create or remove an access group to control NTP access and applies a basic IP access list using the command [**no**] **ntp access-group match-all** | {{ **peer** | **serve** | **serve-only** | **query-only** } *access-list-name* }

Example:

```
switch(config)# ntp access-group match-all
 switch(config)# ntp access-group peer peer-acl
 switch(config)# ntp access-group serve serve-acl
```

ACL processing stops and does not continue to the next access group option if NTP matches a deny ACL rule in a configured peer.

- The **peer** keyword enables the device to receive time requests and NTP control queries and to synchronize itself to the servers specified in the access list.
- The **serve** keyword enables the device to receive time requests and NTP control queries from the servers specified in the access list but not to synchronize itself to the specified servers.
- The **serve-only** keyword enables the device to receive only time requests from servers specified in the access list.
- The **query-only** keyword enables the device to receive only NTP control queries from the servers specified in the access list.
- The **match-all** keyword enables the access group options to be scanned in the following order, from least restrictive to most restrictive: peer, serve, serve-only, query-only. If the incoming packet does not match the ACL in the peer access group, it goes to the serve access group to be processed. If the packet does not match the ACL in the serve access group, it goes to the serve-only access group, and so on.



Note The **match-all** keyword is available beginning with Cisco NX-OS Release 7.0(3)I6(1) and is supported on Cisco Nexus 9000 Series switches and the Cisco Nexus 3164Q, 31128PQ, 3232C, and 3264Q switches.

- The *access-list-name* variable is the name of the NTP access group. The name can be an alphanumeric string up to 64 characters, including special characters.

Step 3 Display the NTP access group configuration using the command **show ntp access-groups**

Example:

```
switch(config)# show ntp access-groups
```

This is an optional step.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure Source IP Address

NTP sets the source IP address for all NTP packets based on the address of the interface through which the NTP packets are sent. You can configure NTP to use a specific source IP address.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Configure the source IP address for all NTP packets using the command [**no**] **ntp source** *ip-address*

Example:

```
switch(config)# ntp source 192.0.2.1
```

The *ip-address* can be in IPv4 or IPv6 format.

Step 3 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure Source Interface

You can configure NTP to use a specific interface.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure the source interface for all NTP packets using the command [**no**] **ntp source-interface** *interface*

Example:

```
switch(config)# ntp source-interface ethernet 2/1
```

Use the **?** keyword to display a list of supported interfaces.

Step 3 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure Logging

You can configure NTP logging in order to generate system logs with significant NTP events. NTP logging is disabled by default.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Enable or disable system logs to be generated with significant NTP events using the command [**no**] **ntp logging**

Example:

```
switch(config)# ntp logging
```

NTP logging is disabled by default.

Step 3 Display the NTP logging configuration status using the command **show ntp logging-status**

Example:

```
switch(config)# show ntp logging-status
```

This is an optional step.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

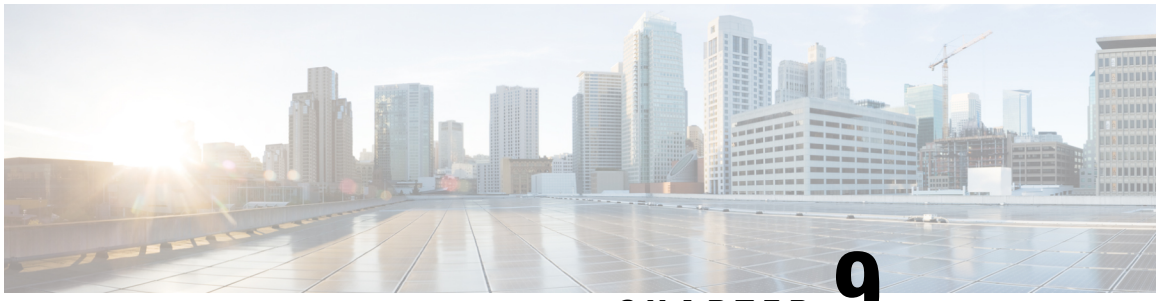
```
switch(config)# copy running-config startup-config
```

This is an optional step.

Additional References

Related Documents

- Clock manager : Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide
- MIBs related to NTP : To locate and download supported MIBs, go to the following URL: https://cisco.github.io/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html



CHAPTER 9

Configuring GNSS

This chapter describes how to configure the Global Navigation Satellite System (GNSS) on Cisco NX-OS devices.

This chapter includes the following sections:

- Global Navigation Satellite System, on page 143
- Guidelines and limitations for GNSS, on page 144
- Enable the GNSS receiver, on page 144
- Commands for verifying GNSS configuration, on page 146

Global Navigation Satellite System

The Global Navigation Satellite System (GNSS) are worldwide navigation infrastructures that

- provide multi-constellation support for receiving,
- offer advanced timing and location integrity features, and
- ensure compatibility with a wide range of satellite services through specific PRN ranges.

Supported GNSS constellations and PRN ranges

The Global Navigation Satellite System (GNSS) receiver is designed to operate on the GPS, Galileo, GLONASS, BeiDou and QZSS L1 frequencies 1551MHz to 1614MHz, standard position service, and Coarse Acquisition code.

When connected to an external GNSS antenna, the receiver contains all the circuitry necessary to automatically acquire GNSS satellite signals, track up to 32 GNSS satellites, and compute location, speed, heading, and time. It provides an accurate one pulse-per-second (PPS) and stable 10-MHz frequency output for internal system use.

It automatically initiates a self-survey upon acquisition of GNSS satellites. When the survey is completed, the receiver switches into the `Over-Determined` timing mode. In this mode, the reference position from the self-survey is maintained in memory and the receiver solves only for clock error and clock bias. The receiver provides for both Position and Time Receiver Autonomous Integrity Monitoring (T-RAIM) which allows the receiver to self-determine position change or to remove a satellite providing incorrect information to the timing solution.

Note No signals are transmitted out of GNSS ports (Antenna port).

The Global Navigation Satellite System (GNSS) supports various satellite constellations, each with specific Pseudo-Random Noise (PRN) numbers. The supported PRN ranges for different GNSS systems are:

- GPS – Supports PRN 1 through 32 only
- GLONASS – Supports PRN 65 through 96 only
- BeiDou – Supports PRN 201 through 237 only
- Galileo – Supports PRN 97 through 133 only
- QZSS – Supports PRN 193 through 195 only
- SBAS – Supports PRN 33 through 54 only

Guidelines and limitations for GNSS

GNSS has these guidelines and limitations.

- Beginning with NX-OS Release 10.3(2)F, the GNSS is supported on the Nexus 93180YC-FX3 switch.

Enable the GNSS receiver

Use this procedure to enable the GNSS Receiver as input.

Procedure

Step 1 Enter global configuration mode and then enable feature frequency-synchronization using the **configure terminal** and **feature frequency-synchronization** commands.

Example:

```
switch# configure terminal
switch(config)# feature frequency-synchronization
switch(config)#
```

Step 2 Enable the GNSS Receiver as input using the **gnss-receiver sync 1/2** command.

Example:

```
switch(config)# gnss-receiver sync 1/2
switch(config-gnss-if)# no shutdown
switch(config-gnss-if)#
```

Step 3 Configure the frequency synchronization on GNSS Receiver using the **frequency synchronization** command.

Example:

```
switch(config-gnss-if)# frequency synchronization
switch(config-gnss-freqsync)# selection input
switch(config-gnss-freqsync)# wait-to-restore 0
switch(config-gnss-freqsync)# exit
```

Step 4 (Optional) Configure the GNSS module to automatically track any satellite or configure it to explicitly use a specific constellation using the **constellation** <*type*> command. The default constellation configuration is **AUTO**.

A list of specific constellations include:

- Auto
- GPS
- GALILEO
- BEIDOU
- QZSS
- GLONASS

Example:

```
switch(config-gnss)# constellation gps
switch(config-gnss)#
```

Step 5 (Optional) Configure the delay that is introduced by the antenna cable using the **cable-delay compensation** command.

For long cable runs, this delay can be significant. The range is –1000000, +1000000 nanoseconds.

Note The cable delay is about 5.9 nanoseconds per meter of cable.

Example:

```
switch(config-gnss)# cable-delay compensation 0
switch(onfig-gnss)#
```

Step 6 (Optional) Configure the required thresholds.

a) (Optional) Configure the lowest satellite elevation for fixes using the **elevation threshold** command.

The range is from 0 to 90°. The default value is 10°.

Example:

```
switch(config-gnss)# elevation threshold 10
switch(onfig-gnss)#
```

b) (Optional) Configure the Carrier to Noise ratio (C/No) using the **snr threshold** command.

The range is 0 to 15db-Hz. The default value is 0.0 C/No.

Note Applicable only in Over-Determined Clock. Not applicable during selfsurvey.

Example:

```
switch(config-gnss)# snr threshold 10
switch(onfig-gnss)#
```

c) (Optional) Configure the Position Dilution of Precision (PDOP) threshold using the **pdop threshold** command.

PDOP indicates the confidence level of a position fix. Low DOP values indicate a high confidence level, while high DOP values indicate a low confidence level. The range is from 0 to 10. The default value is 6.

Example:

```
switch(config-gnss)# pdop threshold 6
switch(onfig-gnss)#
```

Step 7 (Optional) Configure the polarity of the GNSS receiver using the **1pps polarity** command.

It can be set to either positive or negative. The default option is positive.

Example:

```
switch(config-gnss)# 1pps polarity positive
switch(onfig-gnss)#
```

Step 8 (Optional) Enable or disable the jamming occurred when the receiver function is disturbed by external RF sources that interfere with GNSS signals or saturate the antenna LNA or receiver front end using the **anti-jam disable** command. Values are Enable or Disable.

Note If enabled, minimum of two satellites required for fix in Over-Determined Clock mode.

Example:

```
switch(config-gnss)# anti-jam disable
switch(onfig-gnss)#
```

Commands for verifying GNSS configuration

After performing the GNSS configuration tasks, use this reference to check for configuration errors and verify the configuration.

show gnss-receiver

The output of this command displays details of GNSS configuration.

This is an example of the output of the **show gnss-receiver** command.

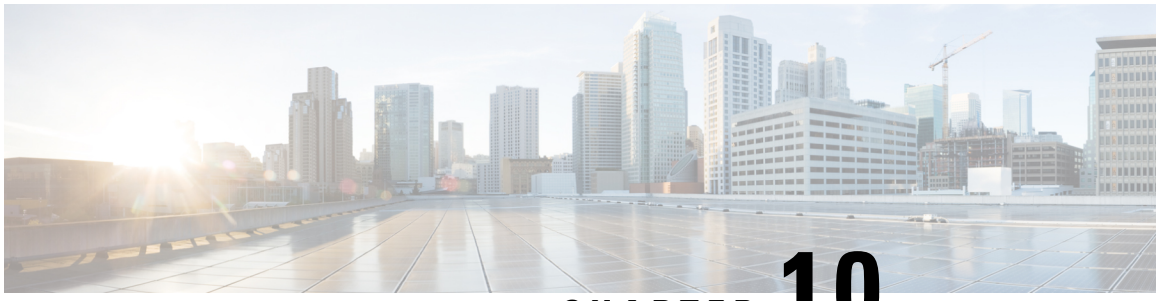
```
switch(config-gnss-if)# sh gnss-receiver
GNSS-receiver SYNC 01/02
Status: Available, Up
Position: 41:11:660 N 74:0:0 W -12.805 m
Time: 1648046422 (UTC offset: 18s)
Firmware version: 1.7
Lock Status: Phase Locked, Receiver Mode: 3D-fix
Survey Progress: 100, Holdover Duration: 0
Major Alarm: Not used
Minor Alarm: Antenna shorted
Anti-jam: Disabled, Cable-delay compensation: 1000
1PPS polarity: Negative
PDOP: 0.000, HDOP: 0.000, VDOP: 0.000, TDOP: 0.001
Constellation: Auto, Satellite Count: 7
```

```
Satellite Thresholds:
SNR - 0 dB-Hz, Elevation - 0 degrees, PDOP - 5, TRAIM - 1 us
Satellite Info:
PRN   Channel Acquisition Ephemeris  SV     Signal
No.   No.     Flag        Flag       Type   Strength   Elevat'n  Azimuth
--- ------- ----------- --------- ----------- -------- -------- --------
11     0      On          On         GPS     0.036     0.076      0.024
28     1      On          On         GPS     0.036     0.025      0.272
1      2      On          On         GPS     0.037     0.089      0.002
19     3      On          On         GPS     0.037     0.036      0.151
14     5      On          On         GPS     0.036     0.019      0.045
17     6      On          On         GPS     0.037     0.025      0.314
23     7      On          On         GPS     0.037     0.014      0.178
switch(config-gnss-if)#
```

show frequency synchronization selection

This is an example of the output of the **show frequency synchronization selection** command.

```
switch(config-gnss-if)# sh frequency synchronization selection
==============
Selection point: System Clock (T0) Selector (2 inputs, 1 selected)
Last programmed 00:53:56 ago, and selection made 00:53:35 ago
Next selection points
Node scoped :
Uses frequency selection
Used for local line interface output
S Input Last Selection Point QL Pri Status
== ======================== ======================== ===== === ===========
Internal0[1] n/a SEC 255 Available
11 GNSS2[1] n/a PRC 100 Locked
==============
Selection point: IEEE 1588 Clock Selector (2 inputs, 1 selected)
Last programmed 00:53:56 ago, and selection made 00:53:55 ago
Next selection points
Node scoped :
Uses frequency selection
S Input Last Selection Point QL Pri Status
== ======================== ======================== ===== === ===========
21 Internal0[1] n/a SEC 255 Holdover
GNSS2[1] n/a PRC 100 Unmonitored
==============
switch(config-gnss-if)#
```



CHAPTER 10

Cisco Discovery Protocol

The Cisco Discovery Protocol (CDP) is a media-independent and protocol-independent protocol that runs on all Cisco-manufactured equipment including routers, bridges, access and communication servers, and switches. You can use CDP to discover and view information about all the Cisco devices that are directly attached to the device.

- Overview, on page 149
- Guidelines and Limitations, on page 151
- Enable or Disable Globally, on page 151
- Enable or Disable on an Interface, on page 153
- Configure Optional CDP Parameters, on page 154

Overview

The Cisco Discovery Protocol (CDP) is a protocol that enables Cisco devices to discover and share information about directly connected devices at the data link layer.

- CDP gathers protocol addresses and platform information from neighboring devices.
- CDP operates at the data link layer and allows devices supporting different Layer 3 protocols to learn about each other.
- CDP periodically advertises device information, including addresses for SNMP messages and hold-time information.

Features and Operation

CDP runs over the data link layer and enables devices to exchange information even if they support different Layer 3 protocols. Each device configured for CDP sends periodic advertisements to a multicast address, advertising at least one address for SNMP messages and hold-time information. The advertisement and hold timers are configurable.

High Availability - Cisco NX-OS supports both stateful and stateless restarts and switchover for CDP. For more information on high availability, see the *Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide*.

Virtualization Support - Cisco NX-OS supports one instance of CDP.

CDP advertises the following type-length-value fields (TLVs):

- Device ID
- Address
- Port ID
- Capabilities
- Version
- Platform
- Native VLAN
- Full or Half Duplex
- MTU
- SysName
- SysObjectID
- Management Address
- Physical Location
- VTP

CDP Version-2 (CDPv2) allows you to track instances where the native VLAN ID or port duplex states do not match between connecting devices.

All CDP packets include a VLAN ID. If you configure CDP on a Layer 2 access port, the CDP packets sent from that access port include the access port VLAN ID. If you configure CDP on a Layer 2 trunk port, the CDP packets sent from that trunk port include the lowest configured VLAN ID allowed on that trunk port. The trunk port can receive CDP packets that include any VLAN ID in the allowed VLAN list for that trunk port. For more information on VLANs, see the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide.

CDP sends the VLAN Trunking Protocol (VTP) type-length-value field (TLV) if the following conditions are met:

- CDP Version 2 is enabled.
- The VTP feature is enabled.
- A VTP domain name is configured.

You can view the VTP information with the **show cdp neighbors detail** command.

Default Settings

This table lists the default settings for CDP parameters.

Parameters	Default
CDP	Enabled globally and on all interfaces
CDP version	Version 2
CDP device ID	Serial number

Parameters	Default
CDP timer	60 seconds
CDP hold timer	180 seconds

Guidelines and Limitations

- CDP can discover up to 256 neighbors per port if the port is connected to a hub with 256 connections.
- CDP must be enabled on the device or you cannot enable it on any interfaces.
- You can configure CDP on physical interfaces and port channels only.

Release Specific Guidelines

Beginning with Cisco NX-OS Release 10.3.1(F)

- CDP is supported on the Cisco Nexus 9808 platform switches.

Beginning with Cisco NX-OS Release 10.4(1)F

- CDP is supported on the following line cards and switches: Cisco Nexus 9804 platform switch, Cisco Nexus X98900CD-A line card, and Cisco Nexus X98900CD-A line card with Cisco Nexus 9808 and 9804 switches

Beginning with Cisco NX-OS Release 10.4(2)F

- CDP is supported on Cisco Nexus 9232E-B1 platform switches.

Enable or Disable Globally

CDP is enabled by default. You can disable CDP and then reenable it.

You must enable CDP on the device before you enable CDP on any interfaces. If CDP is disabled globally and you enable CDP on specified interfaces, CDP will not be active on those interfaces; the system does not return an error message.

Note The Cisco NX-OS commands for this feature may differ from those commands that are used in Cisco IOS.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal switch(config)#
```

Step 2 Enable or disable the CDP feature on the entire device using the command [**no**] **cdp enable**

Example:

```
switch(config)# cdp enable
```

It is enabled by default.

Step 3 (Optional) Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

This example shows how to enable the CDP feature and configure the refresh and hold timers:

```
configure terminal
cdp enable
cdp timer 50
cdp holdtime 100
```

What to do next

To display the CDP configuration, perform one of the following tasks:

Command	Purpose
show cdp all	Displays all interfaces that have CDP enabled.
show cdp entry {**all** | **name** *entry-name*}	Displays the CDP database entries.
show cdp global	Displays the CDP global parameters.
show cdp interface *interface slot*/*port*	Displays the CDP interface status.
show cdp neighbors {**device-id** | **interface** *interface slot*/*port*} [**detail**]	Displays the CDP neighbor status.
show cdp interface *interface slot*/*port*	Displays the CDP traffic statistics on an interface.

Use the **clear cdp counters** command to clear CDP statistics on an interface.

Use the **clear cdp table** command to clear the CDP cache for one or all interfaces.

It is recommended to use the **show cdp neighbors detail** command instead of **show cdp neighbors** command. The **show cdp neighbors** command can display only 13 characters of a platform name. To get the full platform name in the display, use **show cdp neighbors detail** command.

Note When both IPv4 and IPv6 configuration exist under an interface, the last updated IP address, either IPv4 or IPv6, is visible in the output of the **show cdp neighbors** commands. If you need either IPv4 or IPv6 configuration rather than the IP address visible in the output, unconfigure the existing IP configuration and then reconfigure the required IPv4 or IPv6 configuration.

Enable or Disable on an Interface

CDP is enabled by default on an interface. You can disable CDP on an interface.

If CDP is disabled globally and you enable CDP on specified interfaces, CDP will not be active on those interfaces; the system does not return an error message.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Enter interface configuration mode using the command **interface** *interface slot* / *port*

Example:

```
switch(config)# interface ethernet 1/2
 switch(config-if)#
```

Step 3 Enable or disable CDP on this interface using the command [**no**] **cdp enable**

Example:

```
switch(config-if)# cdp enable
```

It is enabled by default.



Note Make sure that CDP is enabled globally on the device.

Step 4 Display CDP information for an interface using the command **show cdp interface** *interface slot* / *port*

Example:

```
switch(config-if)# show cdp interface ethernet 1/2
```

This is an optional step.

Step 5 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure Optional CDP Parameters

You can use the optional commands in this procedure to modify CDP.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal switch(config)#
```

Step 2 Set the CDP version that is supported by the device using the command **cdp advertise** { **v1** | **v2** }

Example:

```
switch(config)# cdp advertise v1
```

This is an optional step. The default is v2.

Step 3 Set the CDP device ID using the command **cdp format device-id** { **mac-address** | **serial-number** | **system-name** }

Example:

```
switch(config)# cdp format device-id mac-address
```

This is an optional step. The options are as follows:

- **mac-address** —The MAC address of the chassis.
- **serial-number** —The chassis serial number/Organizationally Unique Identifier (OUI).
- **system-name** —The system name or fully qualified domain name.

The default is **system-name** .

Step 4 Set the time that CDP holds onto neighbor information before removing it using the command **cdp holdtime** *seconds*

Example:

```
switch(config)# cdp holdtime 150
```

This is an optional step. The range is from 10 to 255 seconds. The default is 180 seconds.

Step 5 Set the refresh time when CDP sends advertisements to neighbors using the command **cdp timer** *seconds*

Example:

```
switch(config)# cdp timer 50
```

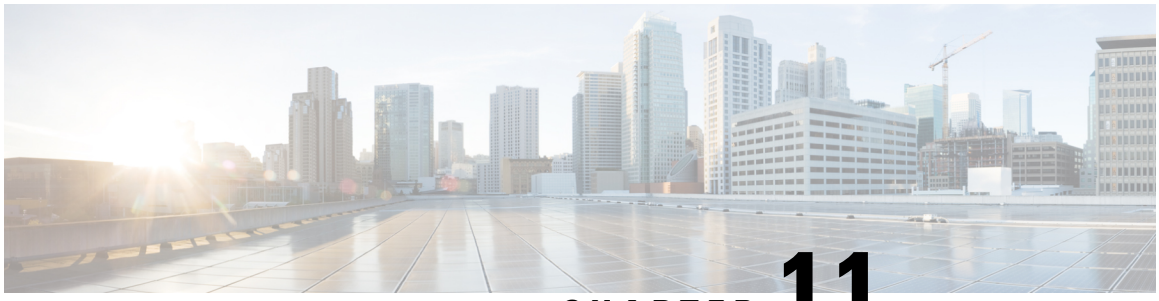
This is an optional step. The range is from 5 to 254 seconds. The default is 60 seconds.

Step 6 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.



CHAPTER 11

System Message Logging

You can use system message logging to control the destination and to filter the severity level of messages that system processes generate. You can configure logging to terminal sessions, a log file, and syslog servers on remote systems.

- Overview, on page 155
- Guidelines and Limitations, on page 157
- Configure System Message Logging to Terminal Sessions, on page 159
- Configure Origin ID for Syslog Messages, on page 161
- Log System Messages to a File, on page 162
- Configure Module and Facility Messages Logging, on page 164
- Configure Logging Syslogs Compliant to RFC 5424, on page 166
- Configure Syslog Servers, on page 167
- Configure Secure Syslog Server, on page 169
- TCP Keepalive for Syslog Logging, on page 170
- Configure CA Certificate, on page 172
- Enroll CA Certificate, on page 173
- Configure Syslog Server on a UNIX or Linux System, on page 174
- Display and Clear Log Files, on page 175
- Configuration for System Message Logging, on page 177
- Related Documents, on page 178

Overview

System message logging is a mechanism that records system events and messages generated by the device, based on RFC 3164 .

- By default, the device outputs messages to terminal sessions and logs system messages to a log file.
- The device logs the most recent 100 messages of severity 0, 1, or 2 to the NVRAM log. Logging to NVRAM cannot be configured.
- You can configure which system messages are logged based on the facility that generated the message and its severity level.

Reference Information

For more information about the system message format and the messages that the device generates, see the Cisco NX-OS System Messages Reference .

The following table describes the severity levels used in system messages. When you configure the severity level, the system outputs messages at that level and lower.

Table 7: System Message Severity Levels

Level	Description
0 – emergency	System unusable
1 – alert	Immediate action needed
2 – critical	Critical condition
3 – error	Error condition
4 – warning	Warning condition
5 – notification	Normal but significant condition
6 – informational	Informational message only
7 – debugging	Appears during debugging only

Syslog Servers

The syslog servers run on remote systems that log system messages based on the syslog protocol. You can configure up to eight IPv4 or IPv6 syslog servers.

To support the same configuration of syslog servers on all switches in a fabric, you can use Cisco Fabric Services (CFS) to distribute the syslog server configuration.

Note When the device first initializes, messages are sent to syslog servers only after the network is initialized.

Secure Syslog Servers

Beginning with Cisco NX-OS Release 9.2(1), you can configure the syslog server with support for a secure TLS transport connectivity to remote logging servers. Additionally, you can enforce the NX-OS switches (client) identity via the mutual authentication configuration. For NX-OS switches, this feature supports TLSv1.1 and TLSv1.2.

The Secure syslog server feature uses the TCP/TLS transport and security protocols to provide device authentication and encryption. This feature enables a Cisco NX-OS device (acting as a client) to make a secure, encrypted outbound connection to remote syslog servers (acting as a server) supporting secure connectivity for logging. With authentication and encryption, this feature allows for a secure communication over an insecure network.

Default Settings

The following table lists the default settings for the system message logging parameters.

Table 8: Default System Message Logging Parameters

Parameters	Default
Console logging	Enabled at severity level 2
Monitor logging	Enabled at severity level 5
Log file logging	Enabled to log messages at severity level 5
Module logging	Enabled at severity level 5
Facility logging	Enabled
Time-stamp units	Seconds
Syslog server logging	Disabled
Syslog server configuration distribution	Disabled

Guidelines and Limitations

System message logging has the following configuration guidelines and limitations:

General Guidelines

- System messages are logged to the console and the log file by default.
- Any system messages that are printed before the syslog server is reachable (such as supervisor active or online messages) cannot be sent to the syslog server.
- Cisco recommends maintaining the logging levels for all processes at default. Increasing the levels to higher values can result in seeing syslog messages that are not intended for customers, can generate false alarms, and are generally supposed to be used for short-term troubleshooting purposes by TAC. Cisco does not provide support for syslog messages at levels above default.
- Due to limitations in Syslog, securePOAP pem file name characters length is limited to 230 characters, though secure POAP supports 256 characters length for a pem file name.
- The maximum length of syslog messages can be 324 characters. If a message exceeds this limit, NX-OS truncates the displayed message. Truncation does not indicate that the underlying operation failed.
- For the secure syslog server(s) to be reachable over an in-band (nonmanagement) interface, the CoPP profile may need tweaks. Especially when multiple logging servers are configured and when many syslogs are generated in a short time (such as, boot up and config application).
- This guideline applies to the user-defined persistent logging file:

  The syslog command, **logging logfile** , allows the configuration of the logfile both in persistent (/logflash/log) and non-persistent locations (/log).

The default logfile is named “messages” and this file, along with backup files (if present) messages.1, messages.2, messages.3, messages.4 cannot be deleted, even by the **delete /log/** or **delete logflash:/log/** commands.

There is a provision to configure custom-named logfiles (**logging logfile** *file-name severity*), however this custom-named file can be deleted by the delete operation. If this occurs, syslog logging does not function.

For example, the custom-named logfile is configured and the same file gets deleted via delete operation. Because this is an intentional delete operation, in order to log the syslog messages on the custom logfiles, you must reconfigure the custom logfile using command **logging logfile** *file-name severity* . Until this configuration is performed, the syslog logging cannot occur.

- Generally, the syslogs display the local time zone. However, few components such as NGINX display the logs in UTC time zone.

Release Specific Guidelines

Beginning with Cisco NX-OS Release 9.2(1)

- You can configure the syslog server with support for a secure TLS transport connectivity to remote logging servers. This feature supports TLS v1.1 and TLS v1.2.

Beginning with Cisco NX-OS Release 10.2(4)M

- TLS v1.3 is supported for syslog on Cisco Nexus 9000 series platform switches.

Beginning with Cisco NX-OS Release 10.4(3)F

- TLS v1.2 and TLS v1.3 is supported for syslog on Cisco Nexus 9000 Series platform switches. TLS v1.1 and TLS v1.0 support for syslog is deprecated.

Beginning with Cisco NX-OS Release 10.3(4a)M

- The existing **logging rfc-strict 5424** command (optional) that enables the syslog protocol RFC 5424 is enhanced by adding a new keyword (**full**) as follows:

  logging rfc-strict 5424 full

  The addition of this keyword ensures complete compliance with the RFC 5424 standard for Syslog Protocol. However, if the values are not available for the `[APP-NAME] [PROCID] [MSG-ID] [STRUCTRED-DATA]` fields, then the nil value is indicated by a dash (-).

Beginning with Cisco NX-OS Release 10.5(3)

- The existing **logging rfc-strict 5424** command (optional) that enables the syslog protocol RFC 5424 is enhanced by adding a new keyword (**utc**) as follows:

  logging rfc-strict 5424 utc

  The addition of this keyword enables the RFC 5424 standard for Syslog Protocol with UTC time format.

  You can also ensure complete compliance with the RFC 5424 standard for Syslog Protocol in UTC time format with the following command: **logging rfc-strict 5424 utc full** .

Starting with Cisco NX-OS Release 10.6(3), console logging is handled asynchronously so that syslogd continues processing and delivering log messages without waiting on console output. This helps ensure that slow console output does not slow down other logging destinations or system operations.

- Use the existing logging drop statistics commands. The output is enhanced to include console-specific drop/error details.

```
show logging dropcount
```

```
clear logging dropcount
```

- Console logging uses a fixed internal buffer (300 messages). If the buffer becomes full, additional console messages can be dropped instead of slowing down syslogd.
- To maintain responsiveness, syslogd does not retry the same console message after a console write failure.

Configure System Message Logging to Terminal Sessions

You can configure the device to log messages by their severity level to console, Telnet, and SSH sessions.

By default, logging is enabled for terminal sessions.



Note The current critical (default) logging level is maintained if the console baud speed is 9600 baud (default). All attempts to change the console logging level will generate an error message. To increase the logging level (above critical), you must change the console baud speed to 38400 baud.



Note Be aware that the Cisco NX-OS commands for this feature might differ from those commands used in Cisco IOS.

Procedure

Step 1 Enable the device to log messages to the console using the command **terminal monitor**

Example:

```
switch# terminal monitor
```

Step 2 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 3 Configure the device to log messages to the console session based on a specified severity level or higher using the command [**no**] **logging console** [*severity-level*]

Example:

```
switch(config)# logging console 3
```

A lower number indicates a higher severity level. Severity levels range from 0 to 7:

- 0 – emergency
- 1 – alert
- 2 – critical
- 3 – error
- 4 – warning
- 5 – notification
- 6 – informational
- 7 – debugging

If the severity level is not specified, the default of 2 is used. The **no** option disables the device's ability to log messages to the console.

Step 4 Display the console logging configuration using the command **show logging console**

Example:

```
switch(config)# show logging console
```

This is an optional step.

Step 5 Enable the device to log messages to the monitor based on a specified severity level or higher using the command [**no**] **logging monitor** [*severity-level*]

Example:

```
switch(config)# logging monitor 3
```

A lower number indicates a higher severity level. Severity levels range from 0 to 7:

- 0 – emergency
- 1 – alert
- 2 – critical
- 3 – error
- 4 – warning
- 5 – notification
- 6 – informational
- 7 – debugging

The configuration applies to Telnet and SSH sessions.

If the severity level is not specified, the default of 2 is used. The **no** option disables the device's ability to log messages to the Telnet and SSH sessions.

Step 6 (Optional) Display the monitor logging configuration using the command **show logging monitor**

Example:

```
switch(config)# show logging monitor
```

This is an optional step.

Step 7 Add the description for physical Ethernet interfaces and subinterfaces in the system message log using the command [**no**] **logging message interface type ethernet description**

Example:

```
switch(config)# logging message interface type ethernet description
```

The description is the same description that was configured on the interface.

The **no** option disables the printing of the interface description in the system message log for physical Ethernet interfaces.

Step 8 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure Origin ID for Syslog Messages

You can configure Cisco NX-OS to append the hostname, an IP address, or a text string to syslog messages that are sent to remote syslog servers.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Specify the hostname, IP address, or text string to be appended to syslog messages that are sent to remote syslog servers using the command **logging origin-id** { **hostname** | **ip** *ip-address* | **string** *text-string* }

Example:

```
switch(config)# logging origin-id string n9k-switch-abc
```

Step 3 Display the configured hostname, IP address, or text string that is appended to syslog messages that are sent to remote syslog servers using the command **show logging origin-id**

Example:

```
switch(config)# show logging origin-id
 Logging origin_id : enabled (string: n9k-switch-abc)
```

This is an optional step.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Log System Messages to a File

You can configure the device to log system messages to a file. By default, system messages are logged to the file /logflash/log/ *logfilename* .

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Configure the nonpersistent or persistent log file parameters using the command [**no**] **logging logfile** *logfile-name severity-level* [**persistent threshold** *percent* | **size** *bytes*]

Example:

```
switch(config)# logging logfile my_log 6

switch(config)# logging logfile my_log 6 persistent threshold 90
```

logfile-name : Configures the name of the log file that is used to store system messages. Default filename is "message".

severity-level : Configures the minimum severity level to log. A lower number indicates a higher severity level. Default is 5. Range is from 0 through 7:

- 0 – emergency
- 1 – alert
- 2 – critical
- 3 – error
- 4 – warning
- 5 – notification
- 6 – informational
- 7 – debugging

persistent threshold *percent* : Optionally configure the threshold percentage for the persistent log file. Range is from 0 through 99.

Note Setting **persistent threshold** to 0 (zero) disables the persistent threshold feature and generates no threshold syslogs.

percent configures the percent threshold size of the persistent file. Once the threshold size is reached, an alert notification message is logged. On reaching 100% utilization of the persistent log file, the system sends another syslog message notification. The system then creates a backup file of the existing log file and starts writing into a new log file with the configured threshold percentage applied. In total, the last five backup files are present at most. After five files, the system deletes files based on the oldest modified.

Note Persistent logging is a system-enabled feature. Log files are located here: /logflash/log/[filename].

Outputs of the following show commands support the persistent log file feature:

- **show logging info**
- **show logging**

The outputs include the following persistent logging information:

```
Logging logflash: enabled (Severity: notifications)(threshold percentage: 99)
 Logging logfile: enabled
 Name - messages: Severity - notifications Size - 4194304
```

size *bytes* : Optionally specify maximum file size. Range is from 4096 through 4194304 bytes.

Step 3 Log interface events using the command **logging event** { **link-status** | **trunk-status** } { **enable** | **default** }

Example:

```
switch(config)# logging event link-status default
```

- **link-status** —Logs all UP/DOWN and CHANGE messages.
- **trunk-status** —Logs all TRUNK status messages.
- **enable** —Specifies to enable logging to override the port level configuration.
- **default** —Specifies that the default logging configuration is used by interfaces that are not explicitly configured.

Step 4 Display the logging configuration using the command **show logging info**

Example:

```
switch(config)# show logging info
```

This is an optional step.

Step 5 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure Module and Facility Messages Logging

You can configure the severity level and time-stamp units of messages logged by modules and facilities.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Enable module log messages that have the specified severity level or higher using the command [**no**] **logging module** [*severity-level*]

Example:

```
switch(config)# logging module 3
```

Severity levels range from 0 to 7:

- 0 – emergency
- 1 – alert
- 2 – critical
- 3 – error
- 4 – warning
- 5 – notification
- 6 – informational
- 7 – debugging

If the severity level is not specified, the default of 5 is used. The **no** option disables module log messages.

Step 3 Display the module logging configuration using the command **show logging module**

Example:

```
switch(config)# show logging module
```

This is an optional step.

Step 4 Enable logging messages from the specified facility that have the specified severity level or higher using the command [**no**] **logging level** *facility severity-level*

Example:

```
switch(config)# logging level aaa 2
```

Severity levels range from 0 to 7:

- 0 – emergency
- 1 – alert
- 2 – critical
- 3 – error
- 4 – warning
- 5 – notification
- 6 – informational
- 7 – debugging

To apply the same severity level to all facilities, use the **all** facility. For defaults, see the **show logging level** command.

The **no** option resets the logging severity level for the specified facility to its default level. If you do not specify a facility and severity level, the device resets all facilities to their default levels.

Step 5 Display the logging level configuration and the system default level by facility using the command **show logging level** [*facility*]

Example:

```
switch(config)# show logging level aaa
```

This is an optional step. If you do not specify a facility, the device displays levels for all facilities.

Note In running configurations, the logging level for authpriv is displayed as authpri in releases earlier than 10.4(3)F and as authpriv from release 10.4(3)F.

Step 6 Enable logging of the Ethernet Port Manager link-up/link-down syslog messages at level 3 using the command [**no**] **logging level** *ethpm*

Example:

```
    switch(config)# logging level ethpm ?
    <0-7>      0-emerg;1-alert;2-crit;3-err;4-warn;5-notif;6-inform;7-debug
link-down  Configure logging level for link down syslog messages
link-up    Configure logging level for link up syslog messages

    switch(config)#logging level ethpm link-down ?
    error  ERRORS
notif  NOTICE
    (config)# logging level ethpm link-down error ?

    <CR>
    (config)# logging level ethpm link-down notif ?
    <CR>
    switch(config)#logging level ethpm link-up ?
    error  ERRORS
```

```
notif  NOTICE
     (config)# logging level ethpm link-up error ?

     <CR>
     (config)# logging level ethpm link-up notif ?
     <CR>
```

Use the **no** option to use the default logging level for Ethernet Port Manager syslog messages.

Step 7 Set the logging time-stamp units using the command [**no**] **logging timestamp** { **microseconds** | **milliseconds** | **seconds** }

Example:

```
switch(config)# logging timestamp milliseconds
```

By default, the units are seconds.

Note This command applies to logs that are kept in the switch. It does not apply to the external logging server.

Step 8 Display the logging time-stamp units configured using the command **show logging timestamp**

Example:

```
switch(config)# show logging timestamp
```

This is an optional step.

Step 9 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure Logging Syslogs Compliant to RFC 5424

The command can be modified in the following ways :

- [*no*] **logging rfc-strict 5424**
- **show logging rfc-strict 5424**

Procedure

Step 1 `switch(config)#` [*no*] logging rfc-strict 5424

(optional) Negate a command or set its defaults

Step 2 `switch(config)#` **logging rfc-strict 5424**

Modify message logging facilities and set RFC to which messages should be compliant.

Step 3 `switch(config)#` show logging rfc-strict 5424

Displays the syslogs which will be compliant to RFC 5424

Configure Syslog Servers

Note Cisco recommends that you configure the syslog server to use the management virtual routing and forwarding (VRF) instance. For more information on VRFs, see Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide.

You can configure up to eight syslog servers that reference remote systems where you want to log system messages.

Note Until Cisco NX-OS Release 10.3(2)F, when the user input certain default values, the running-config of logging server commands showed those default values randomly or inconsistently. However, beginning with Cisco NX-OS Release 10.3(2)F, the running config consistently shows only the non-default values.

For example, in earlier releases, for a certain user input, if the running-config showed `logging server 1.1.1.1 port 514 facility local7 use-vrf default` values, from Cisco NX-OS Release 10.3(2)F onwards, for the same input, the running-config shows only `logging server 1.1.1.1` value. Notice that the default value such as the default port, default facility (local7), and the default VRF are not shown in the running-config.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Configure a syslog server at the specified hostname, IPv4, or IPv6 address using the command [**no**] **logging server** *host* [*severity-level* [**use-vrf** *vrf-name*]]

Example:

```
switch(config)# logging server 192.0.2.253
```

Example:

```
switch(config)# logging server 2001::3 5 use-vrf red
```

You can specify logging of messages to a particular syslog server in a VRF by using the **use-vrf** keyword. The **use-vrf** *vrf-name* keyword identifies the default or management values for the VRF name. The default VRF is marked as 'default'. However, the **show-running** command will not list the default VRF. Severity levels range from 0 to 7:

- 0 – emergency
- 1 – alert
- 2 – critical
- 3 – error
- 4 – warning
- 5 – notification
- 6 – informational
- 7 – debugging

The default outgoing facility is local7.

The **no** option removes the logging server for the specified host.

The first example forwards all messages on facility local 7. The second example forwards messages with severity level 5 or lower to the specified IPv6 address in VRF red.

Note After configuring this command, any one of the following server status is displayed:

- **Configured** – Configuration is successful.
- **No errors found** - If the syslog is transmitted to the remote syslog server successfully, this status is displayed.
- **Temporarily unreachable** - If there is a problem with transmission, this status is displayed. However, internally, the system probes the problem with transmission. After a while, when the issue is resolved, the status changes to No errors found.

Step 3 Enable a source interface for the remote syslog server using the command **logging source-interface loopback** *virtual-interface*

Example:

```
switch(config)# logging source-interface loopback 5
```

The range for the *virtual-interface* argument is from 0 to 1023.

Step 4 Display the syslog server configuration using the command **show logging server**

Example:

```
switch(config)# show logging server
```

This is an optional step.

Step 5 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure Secure Syslog Server

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
                      switch(config)#
```

Step 2 Configure a syslog server at the specified hostname or IPv4 or IPv6 address using the command [**no**] **logging server** *host* [*severity-level* [**port** *port-number*][**secure** [**trustpoint client-identity** *trustpoint-name*]][**use-vrf** *vrf-name*]]

Example:

One-Way TLS Authentication (Server Authentication)

To enable one-way TLS authentication, use the secure keyword. The switch verifies the identity of the syslog server using the CA certificates installed on the switch.

```
switch(config)# logging server 192.0.2.253 secure
```

Example:

Mutual TLS Authentication (Two-Way Authentication)

To enable mutual TLS authentication, use the secure trustpoint client-identity trustpoint-name option. The switch and syslog server validate each other's identity. The switch presents the certificate configured in the specified trustpoint to the server.

```
switch(config)# logging server 2001::3 5 secure trustpoint client-identity myCA use-vrf red
```

Note The default destination port for secure TLS connections is 6514. Configure and authenticate the required CA trustpoints before enabling secure logging.

Optionally, you can enforce a mutual authentication by installing the client identity certificate that is signed by any CA and using the trustpoint client-identity option.

The default destination port for a secure TLS connection is 6514.

Step 3 Enable a source interface for the remote syslog server using the command **logging source-interface** *interface name*

Example:

```
switch(config)# logging source-interface lo0
```

This is an optional step.

Step 4 Display the syslog server configuration using the command **show logging server**

Example:

```
switch(config)# show logging server
```

This is an optional step. If the secure option is configured, the output will have an entry with the transport information. By default, the transport is UDP if the secure option is not configured.

Step 5 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

In Cisco NX-OS Release 9.3(8), when the OCSP responder is down, or if there are OCSP signing issues, SSL connection fails as OCSP works in a strict mode. Hence, beginning with Cisco NX-OS Release 9.3(9), the following new command is introduced to allow you to enable or disable strict-mode.

[**no**] **logging secure ocsp strict**

Note By default, the strict-mode is enabled. Use the no form of the command to enable the non-strict-mode.

TCP Keepalive for Syslog Logging

The TCP Keepalive for Syslog Logging feature improves the reliability of syslog message delivery by maintaining active TCP connections to remote syslog servers, even during periods of inactivity.

- Allows administrators to configure a TCP keepalive idle timeout for syslog over TCP connections.
- Automatically sends keepalive probes during idle periods to prevent connection drops caused by network middleboxes (such as firewalls and load balancers).
- Detects and recovers from dropped TCP connections to ensure continuous syslog logging.

This feature is essential for environments where syslog messages are sent to remote servers over TCP and network devices may enforce idle timeout policies.

- Prevents silent loss of syslog messages by keeping the TCP connection alive, ensuring important audit and security logs are not missed.
- Reduces the need for manual intervention by automatically re-establishing connections if they are dropped due to inactivity.
- Provides flexibility for administrators to tune the idle timeout value to suit their network environment, improving integration with existing security policies and middleboxes.

Guidelines and Limitations for TCP Keepalive

Follow these guidelines and limitations when configuring TCP keepalive for syslog logging.

- The configurable idle timeout range is 300 to 7200 seconds. The default idle timeout is 7200 seconds.
- Once the idle timeout is configured, it can only be reverted to the default value (7200 seconds) by using the `no logging tcp keepalive idle-timeout` command. Restarting the device does not reset the idle timeout setting.
- Keepalive probe interval and probe count are fixed at 75 seconds and 9 probes.
- This feature applies only to syslog servers configured with TCP transport.
- Disabling or misconfiguring keepalive can result in undetected loss of syslog messages.
- Feature configuration changes take effect only for new TCP syslog connections. Existing connections may not inherit updated keepalive settings until they are re-established.
- The idle timeout setting is global and applies to all active syslog servers on the switch. You cannot set different idle timeout values for individual syslog servers.

Configure TCP Keepalive for Syslog Logging

Configure TCP keepalive when syslog is sent over TCP to a remote server and there is a risk of idle connections being dropped.

TCP keepalive is needed when network segments (such as firewalls or load balancers) enforce idle timeouts on TCP connections, which could disrupt syslog logging.

Before you begin

Ensure the following before configuring TCP keepalive for syslog logging:

- Syslog servers are configured to use TCP transport.
- Know the desired idle timeout value for your environment.

Follow these steps to configure TCP keepalive for syslog logging.

Procedure

Step 1 Use the `configure terminal command` to enter global configuration mode.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Use the `logging tcp keepalive idle-timeout` **`idle-timeout value`** command to configure the TCP keepalive idle timeout for syslog logging.

Example:

```
switch(config)# logging tcp keepalive idle-timeout 300
```

Replace *idle-timeout value* with the desired idle timeout in seconds. The default value is 7200 seconds.

Note Only integer values within the range of 300 to 7200 are accepted for the idle timeout. Decimal values are not supported.

Step 3 (Optional) Use the `no logging tcp keepalive idle-timeout` command to revert to the default idle timeout.

Example:

```
switch(config)# no logging tcp keepalive idle-timeout
```

This restores the default idle timeout value (7200 seconds).

Step 4 Use the exit command to leave configuration mode.

Example:

```
switch(config)# exit
```

Step 5 Use the `show running-config | grep "idle"` command to verify the configured TCP keepalive idle timeout for syslog logging.

Example:

```
switch(config)#show running-config | grep "idle"
logging tcp idle-timeout 300
```

If the idle timeout is set to a non-default value, it will be displayed in the output.

Step 6 Use the `show running-config all | grep "idle"` command to check the TCP keepalive idle timeout, including the default value.

Example:

```
switch# show running-config all | grep "idle"
logging tcp idle-timeout 7200
```

If you do not configure the idle timeout, the `show running-config | grep "idle"` command does not show any output. To verify the idle timeout value, use `show running-config all | grep "idle"`. This command shows the default value (7200 seconds) if you have not set a different value.

The switch now maintains active TCP connections for syslog logging according to the configured keepalive idle timeout, reducing the risk of log loss due to idle timeouts.

Configure CA Certificate

For the secure syslog feature support, the remote servers must be authenticated via a trustpoint configuration.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Configure a trustpoint using the command [**no**] **crypto ca trustpoint** *trustpoint-name*

Example:

```
switch(config)# crypto ca trustpoint winca
                        switch(config-trustpoint)#
```

Note You must configure the ip domain-name before the trustpoint configuration.

Step 3 Configure a CA certificate for the trustpoint using the command **crypto ca authenticate** *trustpoint-name*

Example:

```
switch(config-trustpoint)# crypto ca authenticate winca
```

Step 4 Display the configured certificate/chain and the associated trustpoint using the command **show crypto ca certificate**

Example:

```
switch(config)# show crypto ca certificates
```

This is an optional step.

Step 5 Copy the running configuration to the startup configuration so that the trustpoint is persistent across the reload of the device using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Enroll CA Certificate

For mutual authentication, where the remote server wants the NX-OS switch (the client) to identify, that the peer authentication is mandatory, this is an additional configuration to enroll the certificate on the switch.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Configure an RSA key pair using the command **crypto key generate rsa label** *key name* **exportable modules 2048**

Example:

```
switch(config-trustpoint)# crypto key generate rsa label myKey exportable modulus 2048
```

By default, the Cisco NX-OS software generates an RSA key using 1024 bits. Repeat step 2.

Step 3 Associate the keypair generated to the trustpoint CA using the command **rsakeypair** *key-name*

Example:

```
switch(config-trustpoint)# rsakeypair myKey
```

Step 4 Configure a CA certificate for the trustpoint using the command **crypto ca trustpoint** *trustpoint-name*

Example:

```
switch(config)# crypto ca authenticate myCA
```

Step 5 Generate an identity certificate of the switch to enroll it to a CA using the command [**no**] **crypto ca enroll** *trustpoint-name*

Example:

```
switch(config)# crypto ca enroll myCA
```

Step 6 Import the identity certificate signed by the CA to the switch using the command **crypto ca import** *trustpoint-name* **certificate**

Example:

```
switch(config-trustpoint)# crypto ca import myCA certificate
```

Step 7 Display the configured certificate or chain and the associated trustpoint using the command **show crypto ca certificates**

Example:

```
switch# show crypto ca certificates
```

This is an optional step.

Step 8 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch# copy running-config startup-config
```

Configure Syslog Server on a UNIX or Linux System

You can configure a syslog server on a UNIX or Linux system by adding the following line to the /etc/syslog.conf file:

```
    facility.level
<five tab characters>
    action
```

The following table describes the syslog fields that you can configure.

Table 9: Syslog fields in syslog.conf

Field	Description
Facility	Creator of the message, which can be auth, authpriv, cron, daemon, kern, lpr, mail, mark, news, syslog, user, local0 through local7, or an asterisk (*) for all. These facility designators allow you to control the destination of messages based on their origin. **Note** Check your configuration before using a local facility.
Level	Minimum severity level at which messages are logged, which can be debug, info, notice, warning, err, crit, alert, emerg, or an asterisk (*) for all. You can use none to disable a facility.
Action	Destination for messages, which can be a filename, a hostname preceded by the at sign (@), a comma-separated list of users, or an asterisk (*) for all logged-in users.

Procedure

Step 1 Log debug messages with the local7 facility in the file /var/log/myfile.log by adding the following line to the /etc/syslog.conf file

Example:

```
debug.local7 var/log/myfile.log
```

Step 2 Create the log file by entering these commands at the shell prompt

Example:

```
$ touch /var/log/myfile.log
 $ chmod 666 /var/log/myfile.log
```

Step 3 Make sure the system message logging daemon reads the new changes by checking myfile.log after entering this command

Example:

```
$ kill -HUP ~cat /etc/syslog.pid~
```

Display and Clear Log Files

You can display or clear messages in the log file and the NVRAM.

Procedure

Step 1 Required: Display the last number of lines in the logging file using the command **show logging last** *number-lines*

Example:

```
switch# show logging last 40
```

You can specify from 1 to 9999 for the last number of lines.

Step 2 Display the messages in the log file that have occurred within the duration entered using the command **show logging logfile duration** *hh:mm:ss*

Example:

```
switch# show logging logfile duration 15:10:0
```

Step 3 Display the sequence number of the last message in the log file using the command **show logging logfile last-index**

Example:

```
switch# show logging logfile last-index
```

Step 4 Display the messages in the log file that have a timestamp within the span entered using the command **show logging logfile** [**start-time** *yyyy mmm dd hh:mm:ss*] [**end-time** *yyyy mmm dd hh:mm:ss*]

Example:

```
switch# show logging logfile start-time 2013 oct 1 15:10:0
```

If you do not enter an end time, the current time is used. You enter three characters for the month time field and digits for the year and day time fields.

Step 5 Display messages occurring within a range of sequence numbers using the command **show logging logfile** [**start-seqn** *number*] [**end-seqn** *number*]

Example:

```
switch# show logging logfile start-seqn 100 end-seqn 400
```

If you do not include an end sequence number, the system displays messages from the start number to the last message in the log file.

Step 6 Display the messages in the NVRAM using the command **show logging nvram** [**last** *number-lines*]

Example:

```
switch# show logging nvram last 10
```

To limit the number of lines displayed, you can enter the last number of lines to display. You can specify from 1 to 100 for the last number of lines.

Step 7 Clear the contents of the log file using the command **clear logging logfile** [**persistent**]

Example:

```
switch# clear logging logfile
```

persistent : Clears the contents of the log file from the persistent location.

Step 8 Clear the logged messages in NVRAM using the command **clear logging nvram**

Example:

```
switch# clear logging nvram
```

Configuration for System Message Logging

System Message Logging Configuration Example

System message logging configuration enables you to control where and how system messages are logged on the device.

This example shows how to configure system message logging:

```
configure terminal
    logging console 3
    logging monitor 3
    logging logfile my_log 6
    logging module 3
    logging level aaa 2
    logging timestamp milliseconds
    logging server 172.28.254.253
    logging server 172.28.254.254 5 facility local3
    copy running-config startup-config
```

Repeated System Logging Messages

System processes generate logging messages. Depending on the filters used to control which severity levels are generated, a large number of messages can be produced with many of them being repeated.

To make it easier to develop scripts to manage the volume of logging messages, and to eliminate repeated messages from "flooding" the output of the **show logging log** command, the following method of logging repeated messages is used.

In the old method, when the same message was repeated, the default was to state the number of times it reoccurred in the message:

```
2019 Mar 11 13:42:44 Cisco-customer %PTP-2-PTP_INCORRECT_PACKET_ON_SLAVE:
Incorrect delay response packet received on slave interface Eth1/48 by
2c:5a:0f:ff:fe:51:e9:9f. Source Port Identity is 08:00:11:ff:fe:22:3e:4e. Requesting Port
Identity is 00:1c:73:ff:ff:ee:f6:e5
2019 Mar 11 13:43:15 Cisco-customer last message repeated 242 times
```

The new method simply appends the repeat count to the end of the repeated message:

```
2019 Mar 11 13:42:44 Cisco-customer %PTP-2-PTP_INCORRECT_PACKET_ON_SLAVE:
Incorrect delay response packet received on slave interface Eth1/48 by
2c:5a:0f:ff:fe:51:e9:9f. Source Port Identity is 08:00:11:ff:fe:22:3e:4e. Requesting Port
Identity is 00:1c:73:ff:ff:ee:f6:e5

2019 Mar 11 13:43:15 Cisco-customer %PTP-2-PTP_INCORRECT_PACKET_ON_SLAVE:
Incorrect delay response packet received on slave interface Eth1/48 by
2c:5a:0f:ff:fe:51:e9:9f. Source Port Identity is 08:00:11:ff:fe:22:3e:4e. Requesting Port
Identity is 00:1c:73:ff:ff:ee:f6:e5 (message repeated 242 times)
```

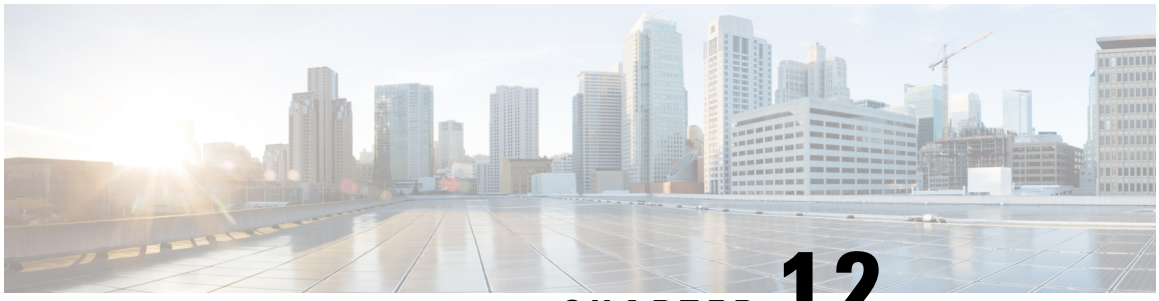
Verify System Message Logging Configuration

To display system message logging configuration information, perform one of the following tasks:

Command	Purpose
show logging console	Displays the console logging configuration.
show logging info	Displays the logging configuration.
show logging last *number-lines*	Displays the last number of lines of the log file.
show logging level [*facility*]	Displays the facility logging severity level configuration.
show logging logfile duration *hh:mm:ss*	Displays the messages in the log file that have occurred within the duration entered.
show logging logfile last-index	Displays the sequence number of the last message in the log file.
show logging logfile [**start-time** *yyyy mmm dd hh:mm:ss*] [**end-time** *yyyy mmm dd hh:mm:ss*]	Displays the messages in the log file based on a start and end date/time.
show logging logfile [**start-seqn** *number*] [**end-seqn** *number*]	Displays messages occurring within a range of sequence numbers. If you do not include an end sequence number, the system displays messages from the start number to the last message in the log file.
show logging module	Displays the module logging configuration.
show logging monitor	Displays the monitor logging configuration.
show logging nvram [**last** *number-lines*]	Displays the messages in the NVRAM log.
show logging server	Displays the syslog server configuration.
show logging timestamp	Displays the logging time-stamp units configuration.

Related Documents

Related Topic	Document Title
Clock manager	Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide



CHAPTER 12

Configure Audit Log Reporting

This chapter describes how to configure audit log reporting on Cisco NX-OS devices.

This chapter contains the following sections:

- AuditD, on page 179
- Guidelines and Limitations, on page 179
- Configure AuditD, on page 180
- Monitor Rules, on page 181
- Verifying AuditD Configuration, on page 182

AuditD

Beginning with Cisco NX-OS Release 10.6(1)F, you can enable AuditD feature to monitor the commands that are executed in guest shell.

Guidelines and Limitations

Following are the guidelines and limitations for AuditD:

- This feature can be enabled only on platforms that have memory of 24GB or more.
- No SNMP Support is available for this feature.
- Tetragon and AuditD features cannot be enabled together. Only one of them can be configured at a time.
- AuditD rules are read-only rules and should not be modified.
- On NX-OS Release10.6(1)F, syslog format for this feature is AUDIT-6-INFO.
- This feature will monitor activities in guest-shell and supervisor only. Monitoring activities on LC or vHost are not supported.
- For audit logs, NX-OS syslog supports maximum syslog size of 1024 bytes (including syslog header). If audit log size exceeds beyond that limit the audit log gets truncated and printed to syslog.
- Beginning with Cisco NX-OS Release 10.6(2)F, these changes are introduced:
    - Syslog format for audit syslog format is AUDIT-6-EVENT.

- Audit log message is rate limited to 1000 message/second. When audit queue size is 1000 messages, any messages generated will get tail dropped by kernel audit.
- In a 4 minute time period, if linux AuditD gets restarted 5 times, it's an error condition and 6th restart within that period is stopped and syslog will be printed:

```
2025 Sep 29 21:29:35 utit_swtele_tb1 %AUDIT-2-AUDITD_EXIT: audit service restarted
more than max restart counts
```

Configure AuditD

Follow the steps to configure AuditD.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Step 2 Use the **feature audit** command to enable AuditD.

Use **No** form of this command to disable the feature.

Step 3 Use the **audit monitor {all | authlog-files | cron-files | dns-client-files | docker | guest-shell | kernel-module-mgmt | process-audit | system-log-files | system-login-reboot | system-software | system-time-change | user-group-config-files | user-privilege-mgmt}** command to enable the rules for AuditD including the rules for Guest Shell.

You cannot configure this command if you have already configured **audit monitor guest-shell** . You need to disable **audit monitor guest-shell** before configuring this command. Use **No** form of this command to delete the configuration.

All - Enable all the rules

guest-shell - Monitor commands executed in guest-shell

From NX-OS 10.6(2)F, following options are introduced:

authlog-files – Monitor authentication log files

cron-files – Monitor cron job files

dns-client-files – Monitor dns client files

docker – Monitor docker

kernel-module-mgmt – Monitor kernel module management

process-audit – Monitor process audit

system-log-files – Monitor system log files

system-login-reboot – Monitor system login and reboot

system-software – Monitor system software

system-time-change – Monitor system time change

user-group-config-files – Monitor user group config files

user-privilege-mgmt – Monitor user privilege mgmt

Step 4 Use the **audit monitor guest-shell** command to enable rules to monitor guest-shell commands.

You cannot configure this command if you have already configured **audit monitor all** . You need to disable **audit monitor all** before configuring this command. Use **No** form of this command to delete the configuration.

Step 5 Use the **logging level audit 6** command to enable AuditD logs print to syslog.

By default, this configuration is disabled. Default logging level is 5. To enable syslog printing for AuditD, apply audit 6. This is an existing logging level configuration that helps streaming syslog to remote server. Use **No** form of this command to delete the configuration.

Audit logs will be available on the switch in /nxos/tmp/auditd/audit.log . Maximum of 5 files each of 8MB can be created in /nxos/tmp/auditd/, and after that logs will get rotated. It is recommended to get the AuditD logs pushed to syslog server.

Step 6 Use the **logging logfile messages 6 size 4194304 persistent threshold 0** command to stream syslog.

Set the logging level to 6.

Monitor Rules

- Following are the monitor rules for guest-shell:

```
-a always,exit -F arch=b64 -S execve -F
dir=/isan/vdc_1/virtual-instance/guestshell+/rootfs -F key=gShell_Cmds
-a always,exit -F arch=b32 -S execve -F
dir=/isan/vdc_1/virtual-instance/guestshell+/rootfs -F key=gShell_Cmds
-w /isan/vdc_1/virtual-instance/guestshell+/rootfs/etc/crontab -p wa -k
gShell_cron_changes
-w /isan/vdc_1/virtual-instance/guestshell+/rootfs/etc/cron.d -p wa -k gShell_cron_changes
-w /isan/vdc_1/virtual-instance/guestshell+/rootfs/etc/cron.daily -p wa -k
gShell_cron_changes
-w /isan/vdc_1/virtual-instance/guestshell+/rootfs/etc/cron.hourly -p wa -k
gShell_cron_changes
-w /isan/vdc_1/virtual-instance/guestshell+/rootfs/etc/cron.weekly -p wa -k
gShell_cron_changes
-w /isan/vdc_1/virtual-instance/guestshell+/rootfs/etc/cron.monthly -p wa -k
gShell_cron_changes
-w /isan/vdc_1/virtual-instance/guestshell+/rootfs/bin/kmod -p wa -k
gShell_modules_changes
-w /isan/vdc_1/virtual-instance/guestshell+/rootfs/etc/passwd -p wa -k
gShell_passwd_changes
-w /isan/vdc_1/virtual-instance/guestshell+/rootfs/etc/shadow -p wa -k
gShell_shadow_changes
-w /isan/vdc_1/virtual-instance/guestshell+/rootfs/etc/group -p wa -k gShell_group_changes
-w /isan/vdc_1/virtual-instance/guestshell+/rootfs/etc/sudoers -p wa -k
gShell_sudoers_changes
-w /isan/vdc_1/virtual-instance/guestshell+/rootfs/etc/hosts -p wa -k gShell_hosts_changes
-w /isan/vdc_1/virtual-instance/guestshell+/rootfs/etc/resolv.conf -p wa -k
gShell_dns_changes
-w /isan/vdc_1/virtual-instance/guestshell+/rootfs/var/volatile/log -p wa -k
gShell_log_changes
-w /isan/vdc_1/virtual-instance/guestshell+/rootfs/usr/bin -p wa -k gShell_usr_bin_changes
-w /isan/vdc_1/virtual-instance/guestshell+/rootfs/usr/sbin -p wa -k
gShell_usr_sbin_changes
```

- Following are the monitor rules for supervisor (default):

```
w /etc/crontab -p wa -k cron_changes
-w /etc/cron.d -p wa -k cron_changes
-w /etc/cron.daily -p wa -k cron_changes
-w /etc/cron.hourly -p wa -k cron_changes
-w /etc/cron.weekly -p wa -k cron_changes
-w /etc/cron.monthly -p wa -k cron_changes
-w /bin -p wa -k bin_changes
-w /sbin -p wa -k sbin_changes
-w /usr/bin -p wa -k usr_bin_changes
-w /usr/sbin -p wa -k usr_sbin_changes
-w /usr/bin/dockerd -p wa -k docker_daemon
-w /bin/kmod -p x -k modules_changes
-w /etc/passwd -p wa -k passwd_changes
-w /etc/shadow -p wa -k shadow_changes
-w /etc/group -p wa -k group_changes
-w /etc/sudoers -p wa -k sudoers_changes
-w /etc/hosts -p wa -k hosts_changes
-w /etc/resolv.conf -p wa -k dns_changes
-w /etc/localtime -p wa -k time_changes
-w /var/volatile/log/auth.log -p wa -k auth_logs
-w /var/volatile/log/sudo.log -p wa -k sudo_usage
-w /var/volatile/log/wtmp -p wa -k shutdown_reboot
-w /var/volatile/log -p wa -k log_changes
-w /logflash/log -p wa -k log_changes
-w /var/lib/docker -p wa -k docker_storage
-a always,exit -F arch=b64 -S execve -F path=/usr/bin/docker -F key=docker_commands
```

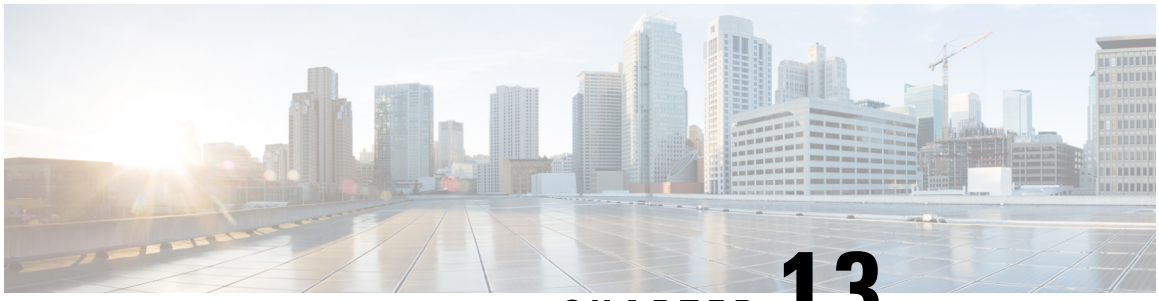
Verifying AuditD Configuration

Use the following commands to see various configuration details about AuditD:

Command	Purpose

show audit status	Displays audit status.
	Example output:
	`switch(config)# show audit status`
	`Backlog: 0`
	`Backlog Limit: 64`
	`Backlog Wait Time: 18000`
	`Enabled: 1`
	`Enabled Timestamp: 2025-Aug-08 21:44:35.278358`
	`Failure: 0`
	`Login UID Immutable: 0 unlocked`
	`Lost: 0`
	`PID: 25426`
	`Rate Limit: 1000`
	`Restart Counts: 0`
	`Restart Timestamp:`
	`switch(config)#`

show running-config audit [all]	Displays the current running configuration for the AuditD feature.
show logging level audit	Displays default logging level and current logging level status.
show tech-support audit	Displays the technical support output for AuditD.



CHAPTER 13

Smart Call Home

Smart Call Home provides an email-based notification for critical system policies. A range of message formats are available for compatibility with pager services, standard email, or XML-based automated parsing applications. You can use this feature to page a network support engineer, email a Network Operations Center, or use Cisco Smart Call Home services to automatically generate a case with the Technical Assistance Center.

- Overview, on page 183
- Guidelines and Limitations, on page 190
- Configuration, on page 191
- Configure Contact Information, on page 194
- Create Destination Profile, on page 195
- Modify Destination Profile, on page 197
- Associate Alert Group with Destination Profile, on page 199
- Add Show Commands to Alert Group, on page 200
- Configure Email Server, on page 201
- Configure VRFs to Send Messages Using HTTP, on page 202
- Configure HTTP Proxy Server, on page 203
- Configure Periodic Inventory Notifications, on page 205
- Disable Duplicate Message Throttling, on page 206
- Enable or Disable Smart Call Home, on page 207
- Configure SMTP-AUTH for Call Home Mail Transfer, on page 208
- Configure Source Interface, on page 210
- Test Smart Call Home Configuration, on page 210
- Additional Reference Information, on page 211

Overview

- Smart Call Home automatically executes relevant CLI commands and attaches their output.
- It provides multiple message format options, including Short Text, Full Text, and XML.
- It supports multiple concurrent message destinations, allowing configuration of up to 50 email addresses per destination profile.

Features

If you have a service contract directly with Cisco, you can register for the Smart Call Home service. Smart Call Home analyzes Smart Call Home messages and provides background information and recommendations. For known issues, particularly online diagnostics failures, Automatic Service Requests are generated with the Cisco TAC.

Smart Call Home offers the following features:

- Continuous device health monitoring and real-time diagnostic alerts.
- Analysis of Smart Call Home messages and, if needed, Automatic Service Request generation, routed to the correct TAC team, including detailed diagnostic information to speed problem resolution.
- Secure message transport directly from your device, through an HTTP proxy server, or a downloadable Transport Gateway (TG). You can use a TG aggregation point to support multiple devices or in cases where security dictates that your devices may not be connected directly to the Internet.
- Web-based access to Smart Call Home messages and recommendations, inventory, and configuration information for all Smart Call Home devices. This feature provides access to associated field notices, security advisories, and end-of-life information.
- Automatic execution and attachment of relevant CLI command output.
- Multiple message format options:
  - Short Text—Suitable for pagers or printed reports.
  - Full Text—Fully formatted message information suitable for human reading.
  - XML—Machine-readable format that uses Extensible Markup Language (XML) and Adaptive Messaging Language (AML) XML schema definition (XSD). The AML XSD is published on the Cisco.com website. The XML format enables communication with the Technical Assistance Center.
- Multiple concurrent message destinations. You can configure up to 50 email destination addresses for each destination profile.
- High Availability - Both stateful and stateless restarts are supported for Smart Call Home.
- Virtualization Support - One instance of Smart Call Home is supported. You can register your contact information at the Smart Call Home web site at the following URL: https://supportforums.cisco.com/community/netpro/solutions/smart_services/smartcallhome. You can test Smart Call Home using the **callhome send** and **callhome test** commands. Smart Call Home is virtual routing and forwarding (VRF) aware. You can configure Smart Call Home to use a particular VRF to reach the Smart Call Home SMTP server.

You need the following information to register:

- The SMARTnet contract number for your device
- Your email address
- Your Cisco.com ID

For more information about Smart Call Home, see the following Smart Call Home page: https://supportforums.cisco.com/community/netpro/solutions/smart_services/smartcallhome

Destination Profiles

A destination profile includes the following information:

- One or more alert groups—The group of alerts that trigger a specific Smart Call Home message if the alert occurs.
- One or more email destinations—The list of recipients for the Smart Call Home messages generated by alert groups assigned to this destination profile.
- Message format—The format for the Smart Call Home message (short text, full text, or XML).
- Message severity level—The Smart Call Home severity level that the alert must meet before Cisco NX-OS generates a Smart Call Home message to all email addresses in the destination profile. Cisco NX-OS does not generate an alert if the Smart Call Home severity level of the alert is lower than the message severity level set for the destination profile.

You can also configure a destination profile to allow periodic inventory update messages by using the inventory alert group that will send out periodic messages daily, weekly, or monthly.

Cisco NX-OS supports the following predefined destination profiles:

- CiscoTAC-1—Supports the Cisco-TAC alert group in XML message format. This profile is preconfigured with the callhome@cisco.com email contact, maximum message size, and message severity level 0. You can change any of the default information for this profile.
- full-text-destination—Supports the full text message format.
- short-text-destination—Supports the short text message format.

Default Settings

This table lists the default settings for Smart Call Home parameters.

Table 10: Default Smart Call Home Parameters

Parameters	Default
Destination message size for a message sent in full text format	2,500,000
Destination message size for a message sent in XML format	2,500,000
Destination message size for a message sent in short text format	4000
SMTP server port number if no port is specified	25
SMTP server priority if no priority is specified	50
Alert group association with profile	All for full-text-destination and short-text-destination profiles. The cisco-tac alert group for the CiscoTAC-1 destination profile.
Format type	XML
Smart Call Home message level	0 (zero)

Parameters	Default
HTTP proxy server use	Disabled and no proxy server configured
Periodic inventory notification (**periodic-inventory notification**)	Enabled by default

Alert Groups

An alert group is a predefined subset of Smart Call Home alerts that are supported in all Cisco Nexus devices. Alert groups allow you to select the set of Smart Call Home alerts that you want to send to a predefined or custom destination profile. The device sends Smart Call Home alerts to email destinations in a destination profile only if that Smart Call Home alert belongs to one of the alert groups associated with that destination profile and if the alert has a Smart Call Home message severity at or above the message severity set in the destination profile.

The following table lists the supported alert groups and the default CLI command output included in Smart Call Home messages generated for the alert group.

Table 11: Alert Groups and Executed Commands

Alert Group	Description	Executed Commands
Cisco-TAC	All critical alerts from the other alert groups destined for Smart Call Home.	Execute commands based on the alert group that originates the alert.
Configuration	Periodic events related to configuration.	**show module** **show version**
Diagnostic	Events generated by diagnostics.	**show diagnostic result module all detail** **show diagnostic result module** *number* **detail** **show hardware** **show logging last 200** **show module** **show sprom all** **show tech-support gold** **show tech-support ha** **show tech-support platform** **show version**

Alert Group	Description	Executed Commands
EEM	Events generated by EEM.	**show diagnostic result module all detail** **show diagnostic result module** *number* **detail** **show module** **show tech-support gold** **show tech-support ha** **show tech-support platform**
Environmental	Events related to power, fan, and environment-sensing elements such as temperature alarms.	**show environment** **show logging last 200** **show module** **show version**
Inventory	Inventory status that is provided whenever a unit is cold booted or when FRUs are inserted or removed. This alert is considered a noncritical event, and the information is used for status and entitlement.	**show inventory** **show license usage** **show module** **show sprom all** **show system uptime** **show version**
License	Events related to licensing and license violations.	**show logging last 200**

Alert Group	Description	Executed Commands
Linecard hardware	Events related to standard or intelligent switching modules.	**show diagnostic result module all detail** **show diagnostic result module** *number* **detail** **show hardware** **show logging last 200** **show module** **show sprom all** **show tech-support ethpm** **show tech-support gold** **show tech-support ha** **show tech-support platform** **show version**
Supervisor hardware	Events related to supervisor modules.	**show diagnostic result module all detail** **show hardware** **show logging last 200** **show module** **show sprom all** **show tech-support ethpm** **show tech-support gold** **show tech-support ha** **show tech-support platform** **show version**
Syslog port group	Events generated by the syslog PORT facility.	**show license usage** **show logging last 200**

Alert Group	Description	Executed Commands
System	Events generated by failure of a software system that is critical to unit operation.	**show diagnostic result module all detail** **show hardware** **show logging last 200** **show module** **show sprom all** **show tech-support ethpm** **show tech-support gold** **show tech-support ha** **show tech-support platform**
Test	User-generated test message.	**show module** **show version**

Smart Call Home maps the syslog severity level to the corresponding Smart Call Home severity level for syslog port group messages.

You can customize predefined alert groups to execute additional CLI **show** commands when specific events occur and send that **show** output with the Smart Call Home message.

You can add **show** commands only to full text and XML destination profiles. Short text destination profiles do not support additional **show** commands because they only allow 128 bytes of text.

Message Levels

Smart Call Home allows you to filter messages based on their level of urgency. You can associate each predefined or user-defined destination profile with a Smart Call Home threshold from 0 (least urgent) to 9 (most urgent). The default is 0 (all messages are sent).

Syslog severity levels are mapped to the Smart Call Home message level.



Note Smart Call Home and Syslogs use different severity levels (see the following table). Smart Call Home does not change the syslog message level in the message text.

The following table lists each Smart Call Home message level keyword and the corresponding syslog level for the syslog port alert group.

Table 12: Severity and Syslog Level Mapping

Smart Call Home Level	Keyword	Syslog Level	Description
9	Catastrophic	N/A	Network-wide catastrophic failure.

Smart Call Home Level	Keyword	Syslog Level	Description
8	Disaster	N/A	Significant network impact.
7	Fatal	Emergency (0)	System is unusable.
6	Critical	Alert (1)	Critical conditions that indicate that immediate attention is needed.
5	Major	Critical (2)	Major conditions.
4	Minor	Error (3)	Minor conditions.
3	Warning	Warning (4)	Warning conditions.
2	Notification	Notice (5)	Basic notification and informational messages. Possibly independently insignificant.
1	Normal	Information (6)	Normal event signifying return to normal state.
0	Debugging	Debug (7)	Debugging messages.

Guidelines and Limitations

Smart Call Home has the following configuration guidelines and limitations:

- If there is no IP connectivity or if the interface in the virtual routing and forwarding (VRF) instance to the profile destination is down, the device cannot send Smart Call Home messages.
- Smart Call Home operates with any SMTP server.
- You can configure up to five SMTP servers for Smart Call Home.
- Link up/down syslog messages do not trigger Smart Call Home messages or alert notifications.
- When configuring Smart Call Home commands such as street address, customer ID, and site ID, you must configure each one of these commands as individual command instead of grouping them with semi-colon separator.
- Callhome does not support specifying a source interface using the **ip http source-interface** command.

Prerequisites

Smart Call Home has the following prerequisites:

- To send messages to an email address, you must first configure an email server. To send messages using HTTP, you must have access to an HTTPS server and have a valid certificate installed on the Cisco Nexus device.
- Your device must have IP connectivity to an email server or HTTPS server.

- You must first configure the contact name (SNMP server contact), phone, and street address information. This step is required to determine the origin of messages received.
- If you use Smart Call Home, you need an active service contract for the device that you are configuring.

Release Specific Limitations

Beginning with Cisco NX-OS Release 10.2(3)F

- SMTP-AUTH is supported for secure call home mail transfer on Cisco Nexus 9000 Series platform switches.

Beginning with Cisco NX-OS Release 10.3(2)F

- You can configure the **source-interface** command under callhome to specify a source interface.

Beginning with Cisco NX-OS Release 10.4(3)F

- Smart Call Home will use only TLSv1.2 and TLSv1.3 for secure email transfer and HTTP transfer on Cisco Nexus 9000 Series switches. TLSv1.1 support for Smart Call Home is deprecated.

Database Merge Guidelines

When you merge two Smart Call Home databases, the following guidelines apply:

- The merged database contains the following information:
  - A superset of all the destination profiles from the merging devices.
  - The destination profile email addresses and alert groups.
  - Other configuration information (for example, message throttling, or periodic inventory) present in the managing device.
- Destination profile names cannot be duplicated within the merging devices—even though the configurations are different, the names cannot be duplicated. If a profile name is duplicated, one of the duplicate profiles must first be deleted or the merger fails.

Configuration

The configuration process involves completing the Smart Call Home configuration procedures in a recommended sequence.

- Follow the recommended order for Smart Call Home configuration steps.
- Some steps are optional and may be performed as needed.
- Cisco NX-OS commands may differ from Cisco IOS commands.

Configuration Sequence

We recommend that you complete the Smart Call Home configuration procedures in the following sequence:

1. Configure Contact Information, on page 194

2. Create Destination Profile, on page 195
3. Associate Alert Group with Destination Profile, on page 199
4. Add Show Commands to Alert Group, on page 200 (Optional)
5. Enable or Disable Smart Call Home, on page 207
6. Configure Source Interface, on page 210 (Optional)
7. Test Smart Call Home Configuration, on page 210 (Optional)



Note Be aware that the Cisco NX-OS commands may differ from the Cisco IOS commands.

Verify Configuration

To display Smart Call Home configuration information, perform one of the following tasks:

Command	Purpose
show callhome	Displays the Smart Call Home configuration.
show callhome destination-profile *name*	Displays one or more Smart Call Home destination profiles.
show callhome transport	Displays the transport-related configuration for Smart Call Home.
show callhome user-def-cmds	Displays CLI commands added to any alert groups.
show running-config callhome [**all**]	Displays the running configuration for Smart Call Home.
show startup-config callhome	Displays the startup configuration for Smart Call Home.
show tech-support callhome	Displays the technical support output for Smart Call Home.

Configuration Examples

This example shows how to create a destination profile called Noc101, associate the Configuration alert group to that profile, configure contact and email information, and specify the VRF used to send Smart Call Home messages over HTTP:

```
configure terminal
snmp-server contact person@company.com
callhome
distribute
email-contact admin@Mycompany.com
phone-contact +1-800-123-4567
streetaddress 123 Anystreet st. Anytown,AnyWhere
destination-profile Noc101 format full-txt
destination-profile full-text-destination email-addr person@company.com
destination-profile full-text-destination message-level 5
destination-profile Noc101 alert-group Configuration
alert-group Configuration user-def-cmd show ip route
transport email mail-server 192.0.2.10 priority 1
transport http use-vrf Blue
```

```
enable
commit
```

This example shows how to configure multiple SMTP servers for Smart Call Home messages:

```
configure terminal
callhome
transport email mail-server 192.0.2.10 priority 4
transport email mail-server 172.21.34.193
transport email smtp-server 10.1.1.174
transport email mail-server 64.72.101.213 priority 60
transport email from person@company.com
transport email reply-to person@company.com
commit
```



Note Configuration of multiple smtp-servers for authentication purpose using the **callhome email mail-server** command is not supported.

Based on the configuration above, the SMTP servers would be tried in this order:

10.1.1.174 (priority 0)

192.0.2.10 (priority 4)

172.21.34.193 (priority 50, which is the default)

64.72.101.213 (priority 60)



Note The **transport email smtp-server** command has a priority of 0, which is the highest. The server specified by this command is tried first followed by the servers specified by the **transport email mail-server** commands in order of priority.

This example shows how to configure Smart Call Home to send HTTP messages through an HTTP proxy server:

```
configure terminal
callhome
transport http proxy server 10.10.10.1 port 4
transport http proxy enable
commit
```

This example shows how to configure SMTP-AUTH servers for call home mail transfer:

```
callhome
email-contact admin@Mycompany.com
destination-profile testProfile-1
destination-profile testProfile-1 format XML
destination-profile testProfile-1 index 1 email-addr person@company.com
destination-profile testProfile-1 alert-group all
destination-profile full_txt alert-group test
transport email from callhome_person@company.com
transport email smtp-server 10.1.1.174 port 587 use-vrf management
transport email username user1 passwd Y2FsbGhvbWUK encrypted
transport http use-vrf management
enable
```

Configure Contact Information

You must configure the email, phone, and street address information for Smart Call Home. You can optionally configure the contract ID, customer ID, site ID, and switch priority information.

You must configure each one of these Smart Call Home commands as individual command instead of grouping them with semi-colon separator.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure the SNMP sysContact using the command **snmp-server contact** *sys-contact*

Example:

```
switch(config)# snmp-server contact personname@companyname.com
```

Step 3 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
 switch(config-callhome)#
```

Step 4 Configure the email address for the person primarily responsible for the device using the command **email-contact** *email-address*

Example:

```
switch(config-callhome)# email-contact admin@Mycompany.com
```

The *email-address* can be up to 255 alphanumeric characters in email address format.



Note You can use any valid email address. The address cannot contain spaces.

Step 5 Configure the phone number in international phone number format for the person primarily responsible for the device using the command **phone-contact** *international-phone-number*

Example:

```
switch(config-callhome)# phone-contact +1-800-123-4567
```

The *international-phone-number* can be up to 17 alphanumeric characters and must be in international phone number format.



Note The phone number cannot contain spaces. Use the plus (+) prefix before the number.

Step 6 Configure the street address as an alphanumeric string with white spaces for the person primarily responsible for the device using the command **streetaddress** *address*

Example:

```
switch(config-callhome)# streetaddress 123 Anystreet st. Anytown,AnyWhere
```

The *address* can be up to 255 alphanumeric characters. Spaces are accepted.

Step 7 Configure the contract number for this device from the service agreement using the command **contract-id** *contract-number* and configure the customer number for this device from the service agreement using the command **customer-id** *customer-number*

Example:

```
switch(config-callhome)# contract-id Contract5678
```

Example:

```
switch(config-callhome)# customer-id Customer123456
```

This is an optional step.

The *contract-number* can be up to 255 alphanumeric characters in free format. The *customer-number* can be up to 255 alphanumeric characters in free format.

Step 8 Configure the site number for this device using the command **site-id** *site-number* and configure the switch priority for this device using the command **switch-priority** *number* .

Example:

```
switch(config-callhome)# site-id Site1
```

Example:

```
switch(config-callhome)# switch-priority 3
```

This is an optional step.

The *site-number* can be up to 255 alphanumeric characters in free format. The range is from 0 to 7, with 0 being the highest priority and 7 the lowest. The default is 7.

Step 9 Commit the Smart Call Home configuration commands using the command **commit** and display a summary of the Smart Call Home configuration using the command **show callhome**

Example:

```
switch(config-callhome)# commit
```

Example:

```
switch(config-callhome)# show callhome
```

What to do next

Copy the running configuration to the startup configuration using the command **copy running-config startup-config** and create a destination profile.

Create Destination Profile

You can create a user-defined destination profile and configure its message format.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
 switch(config-callhome)#
```

Step 3 Create a new destination profile using the command **destination-profile** *name*

Example:

```
switch(config-callhome)# destination-profile Noc101
```

The name can be any alphanumeric string up to 31 characters.

Step 4 Set the message format for the profile using the command **destination-profile** *name* **format** { **XML** | **full-txt** | **short-txt** }

Example:

```
switch(config-callhome)# destination-profile Noc101 format full-txt
```

The name can be any alphanumeric string up to 31 characters.

Step 5 Commit the Smart Call Home configuration commands using the command **commit**

Example:

```
switch(config-callhome)# commit
```

Step 6 Display information about one or more destination profiles using the command **show callhome destination-profile** [**profile** *name*]

Example:

```
switch(config-callhome)# show callhome destination-profile profile Noc101
```

This is an optional step.

Step 7 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

What to do next

Associate one or more alert groups with a destination profile.

Modify Destination Profile

You can modify the following attributes for a predefined or user-defined destination profile:

- Destination email address—The actual address, pertinent to the transport mechanism, to which the alert should be sent.
- Destination URL—The HTTP or HTTPS URL that defines where alerts should be sent.
- Transport method—The email or HTTP transport that determines which type of destination addresses are used.
- Message formatting—The message format used for sending the alert (full text, short text, or XML).
- Message level—The Smart Call Home message severity level for this destination profile.
- Message size—The allowed length of a Smart Call Home message sent to the email addresses in this destination profile.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
 switch(config-callhome)#
```

Step 3 Configure an email address for a user-defined or predefined destination profile using the command **destination-profile** { *name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination** } **email-addr** *address*

Example:

```
switch(config-callhome)# destination-profile full-txt-destination email-addr person@place.com
```

You can configure up to 50 email addresses in a destination profile.

Step 4 Configure an HTTP or HTTPS URL for a user-defined or predefined destination profile using the command **destination-profile** { *name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination** } **http** *address*

Example:

```
switch(config-callhome)# destination-profile CiscoTAC-1 http
https://tools.cisco.com/its/service/oddce/services/DDCEService
```

The URL can be up to 255 characters.

Step 5 Configure an email or HTTP transport method for a user-defined or predefined destination profile using the command **destination-profile** { *name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination** } **transport-method** { **email** | **http** }

Example:

```
switch(config-callhome)# destination-profile CiscoTAC-1 transport-method http
```

The type of transport method that you choose determines the configured destination addresses of that type.

Step 6 Configure the Smart Call Home message severity level for this destination profile using the command **destination-profile** { *name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination** } **message-level** *number*

Example:

```
switch(config-callhome)# destination-profile full-txt-destination message-level 5
```

Cisco NX-OS sends only alerts that have a matching or higher Smart Call Home severity level to destinations in this profile. The range is from 0 to 9, where 9 is the highest severity level.

Step 7 Configure the maximum message size for this destination profile using the command **destination-profile** { *name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination** } **message-size** *number*

Example:

```
switch(config-callhome)# destination-profile full-txt-destination message-size 100000
```

The range is from 0 to 5000000. The default is 2500000.

Step 8 Commit the Smart Call Home configuration commands using the command **commit**

Example:

```
switch(config-callhome)# commit
```

Step 9 Display information about one or more destination profiles using the command **show callhome destination-profile** [**profile** *name*] and copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config-callhome)# show callhome destination-profile profile full-text-destination
```

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

What to do next

Associate one or more alert groups with a destination profile.

Associate Alert Group with Destination Profile

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
 switch(config-callhome)#
```

Step 3 Associate an alert group with this destination profile using the command **destination-profile** { *name* | **CiscoTAC-1** | **full-txt-destination** | **short-txt-destination** } **alert-group** { **All** | **Cisco-TAC** | **Configuration** | **Diagnostic** | **EEM** | **Environmental** | **Inventory** | **License** | **Supervisor-Hardware** | **Syslog-group-port** | **System** | **Test** }

Example:

```
switch(config-callhome)# destination-profile Noc101 alert-group All
```

Use the **All** keyword to associate all alert groups with the destination profile.

Step 4 Commit the Smart Call Home configuration commands using the command **commit**

Example:

```
switch(config-callhome)# commit
```

Step 5 Display information about one or more destination profiles using the command **show callhome destination-profile** [**profile** *name*]

Example:

```
switch(config-callhome)# show callhome destination-profile profile Noc101
```

This is an optional step.

Step 6 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

What to do next

Optionally add **show** commands to an alert group and then configure the SMTP email server.

Add Show Commands to Alert Group

You can assign a maximum of five user-defined CLI **show** commands to an alert group.

Note You cannot add user-defined CLI **show** commands to the CiscoTAC-1 destination profile.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
 switch(config-callhome)#
```

Step 3 Add the **show** command output to any Smart Call Home messages sent for this alert group using the command **alert-group** { **Configuration** | **Diagnostic** | **EEM** | **Environmental** | **Inventory** | **License** | **Supervisor-Hardware** | **Syslog-group-port** | **System** | **Test** } **user-def-cmd** *show-cmd*

Example:

```
switch(config-callhome)# alert-group Configuration user-def-cmd show ip route
```

Only valid **show** commands are accepted.

Step 4 Commit the Smart Call Home configuration commands using the command **commit**

Example:

```
switch(config-callhome)# commit
```

Step 5 Display information about all user-defined **show** commands added to alert groups using the command **show callhome user-def-cmds**

Example:

```
switch(config-callhome)# show callhome user-def-cmds
```

This is an optional step.

Step 6 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

What to do next

Configure Smart Call Home to connect to the SMTP email server.

Configure Email Server

You must configure the SMTP server address for the Smart Call Home functionality to work. You can also configure the from and reply-to email addresses.

You can configure up to five SMTP servers for Smart Call Home. The servers are tried based on their priority. The highest priority server is tried first. If the message fails to be sent, the next server in the list is tried until the limit is exhausted. If two servers have equal priority, the one that was configured earlier is tried first.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
  switch(config-callhome)#
```

Step 3 Configure the SMTP server as the domain name server (DNS) name, IPv4 address, or IPv6 address using the command **transport email mail-server** *ip-address* [**port** *number*] [**priority** *number*] [**use-vrf** *vrf-name*]

Example:

```
switch(config-callhome)# transport email mail-server 192.0.2.1 use-vrf Red
```

Optionally configures the port number. The port range is from 1 to 65535. The default port number is 25.

Also optionally configures the priority of the SMTP server. The priority range is from 1 to 100, with 1 being the highest priority and 100 the lowest. If you do not specify a priority, the default value of 50 is used.

Also optionally configures the VRF to use when communicating with this SMTP server. The VRF specified is not used to send messages using HTTP.

Step 4 Configure the email from field for Smart Call Home messages using the command **transport email from** *email-address*

Example:

```
switch(config-callhome)# transport email from person@company.com
```

This is an optional step.

Step 5 Configure the email reply-to field for Smart Call Home messages using the command **transport email reply-to** *email-address*

Example:

```
switch(config-callhome)# transport email reply-to person@company.com
```

This is an optional step.

Step 6 Commit the Smart Call Home configuration commands using the command **commit**

Example:

```
switch(config-callhome)# commit
```

Step 7 Display the transport-related configuration for Smart Call Home using the command **show callhome transport**

Example:

```
switch(config-callhome)# show callhome transport
```

This is an optional step.

Step 8 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

What to do next

Optionally use VRFs to send Smart Call Home messages over HTTP.

Configure VRFs to Send Messages Using HTTP

You can use VRFs to send Smart Call Home messages over HTTP. If HTTP VRFs are not configured, the default VRF is used to transport messages over HTTP.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
 switch(config-callhome)#
```

Step 3 Configure the VRF used to send email and other Smart Call Home messages over HTTP using the command **transport http use-vrf** *vrf-name*

Example:

```
switch(config-callhome)# transport http use-vrf Blue
```

Step 4 Commit the Smart Call Home configuration commands using the command **commit**

Example:

```
switch(config-callhome)# commit
```

Step 5 Display information about Smart Call Home using the command **show callhome**

Example:

```
switch(config-callhome)# show callhome
```

This is an optional step.

Step 6 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

What to do next

Optionally configure Smart Call Home to send HTTP messages through an HTTP proxy server.

Configure HTTP Proxy Server

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
 switch(config-callhome)#
```

Step 3 Configure the HTTP proxy server domain name server (DNS) name, IPv4 address, or IPv6 address using the command **transport http proxy server** *ip-address* [**port** *number*]

Example:

```
switch(config-callhome)# transport http proxy server 192.0.2.1
```

Optionally configures the port number. The port range is from 1 to 65535. The default port number is 8080.

Step 4 Enable Smart Call Home to send all HTTP messages through the HTTP proxy server using the command **transport http proxy enable**

Example:

```
switch(config-callhome)# transport http proxy enable
```

Note You can execute this command only after the proxy server address has been configured.

Note The VRF used for transporting messages through the proxy server is the same as that configured using the **transport http use-vrf** command.

Step 5 Commit the Smart Call Home configuration commands using the command **commit**

Example:

```
switch(config-callhome)# commit
```

Step 6 Display the transport-related configuration for Smart Call Home using the command **show callhome transport**

Example:

```
switch(config-callhome)# show callhome transport
```

This is an optional step.

Step 7 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

What to do next

Optionally configure your device to periodically send inventory notifications.

Configure Periodic Inventory Notifications

You can configure the device to periodically send a message with an inventory of all software services currently enabled and running on the device along with hardware inventory information. The device generates two Smart Call Home notifications: periodic configuration messages and periodic inventory messages.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
  switch(config-callhome)#
```

Step 3 Configure periodic inventory messages using the command **periodic-inventory notification** [**interval** *days*] [**timeofday** *time*]

Example:

```
switch(config-callhome)# periodic-inventory notification interval 20
```

The interval range is from 1 to 30 days, and the default is 7 days. The *time* argument is in HH:MM format. It defines at what time of the day every *X* days an update is sent (where *X* is the update interval).

Step 4 Commit the Smart Call Home configuration commands using the command **commit**

Example:

```
switch(config-callhome)# commit
```

Step 5 Display information about Smart Call Home using the command **show callhome**

Example:

```
switch(config-callhome)# show callhome
```

This is an optional step.

Step 6 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

What to do next

Optionally disable duplicate message throttling.

Disable Duplicate Message Throttling

You can limit the number of duplicate messages received for the same event. By default, the device limits the number of duplicate messages received for the same event. If the number of duplicate messages sent exceeds 30 messages within a 2-hour time frame, the device discards further messages for that alert type.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
  switch(config-callhome)#
```

Step 3 Disable duplicate message throttling for Smart Call Home using the command **no duplicate-message throttle**

Example:

```
switch(config-callhome)# no duplicate-message throttle
```

Duplicate message throttling is enabled by default.

Step 4 Commit the Smart Call Home configuration commands using the command **commit**

Example:

```
switch(config-callhome)# commit
```

Step 5 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

What to do next

Enable Smart Call Home.

Enable or Disable Smart Call Home

Once you have configured the contact information, you can enable the Smart Call Home function.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
  switch(config-callhome)#
```

Step 3 Enable or disable Smart Call Home using the command [**no**] **enable**

Example:

```
switch(config-callhome)# enable
```

Smart Call Home is disabled by default.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

What to do next

Optionally generate a test message.

Configure SMTP-AUTH for Call Home Mail Transfer

You can use the SMTP-AUTH feature for call home mail transfer to share mails in a secure way using standard SMTP-AUTH TCP port 587 or 465, or any other user-defined port, instead of clear text over port 25. This feature is supported from Cisco NX-OS Release 10.2(3)F.

Before you begin

- SMTP-AUTH server certificate should be installed on the switch.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal** and enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch# configure terminal
switch(config)#
```

Example:

```
switch(config)# callhome
switch(config-callhome)#
```

Step 2 Configure the email address for the person primarily responsible for the device using the command **email-contact** *email-address*

Example:

```
switch(config-callhome)# email-contact admin@Mycompany.com
```

The *email-address* can be up to 255 alphanumeric characters in email address format.



Note You can use any valid email address. The address cannot contain spaces.

Step 3 Create a new destination profile using the command **destination-profile** *name* and set the message format for the profile using the command **destination-profile** *name* **format** { **XML** | **full-txt** | **short-txt** }

Example:

```
switch(config-callhome)# destination-profile testProfile-1
```

Example:

```
switch(config-callhome)# destination-profile testProfile-1 format XML
```

The name can be any alphanumeric string up to 31 characters.

Step 4 Configure an email address to which the secure mail must be delivered using the command **destination-profile** *name* **email-address** *email-address* and associate all the alert groups with the destination profile using the command **destination-profile** *name* **alert-group all**

Example:

```
switch(config-callhome)# destination-profile testProfile-1
index 1 email address person@company.com
```

Example:

```
switch(config-callhome)# destination-profile testProfile-1
alert-group all
```

You can configure up to 50 email addresses in a destination profile.

Step 5 Configure the email from **callhome** field for Smart Call Home messages using the command **transport email from callhome_** *email-address*

Example:

```
switch(config)# transport email from callhome_person@company.com
```

Step 6 Enable SMTP-AUTH mail transfer method using the command **transport email smtp-server** *hostname/ip-address* **port 465 use-vrf** *vrf-name*

transport email smtp-server *hostname/ip-address* **port 587 use-vrf** *vrf-name*

Example:

```
switch(config)# transport email smtp-server 10.1.1.174 port 465
use-vrf management
```

```
switch(config)# transport email smtp-server 10.1.1.174 port 587
use-vrf management
```

STARTTLS-based SMTP-AUTH is over the standard TCP ports, that is, 465 and 587 ports.

Step 7 Accept username and password and passes these details for SMTP-AUTH authentication using the command **transport email username** *username* **passwd** *password* { **cleartext** | **encrypted** }

Example:

```
switch(config)# transport email username user1 passwd Y2FsbGhvbWUK encrypted
```

The username should be alphanumeric and must be less than 256 bytes. Password option can be entered in cleartext or encrypted format (if the user already has the encrypted password). The password length must be less than 64 bytes for the cleartext option and less than 256 bytes for the encrypted option.

Note SMTP-AUTH fails in the following scenarios:

- if the password in cleartext is more than 56 characters in length.
- if the password has any of the following special characters: Dollar sign - **$**, Parentheses - **(** and **)**, Ampersand - **&**, Square Brackets - **[** and **]**, Semicolon - **;** , Question mark - **?** , Vertical bar or pipe - **|** , Apostrophe - **'**, Quotation marks - **'** , **"** , **‘** , **’** , **“** , and **”** and Less-than and More-than signs - **>** and **<**

Step 8 Configure the VRF used to send email and other Smart Call Home messages over HTTP using the command **transport http use-vrf** *vrf-name*

Example:

```
switch(config)# transport http use-vrf management
```

This is an optional step.

Step 9 Enable Smart Call Home using the command [**no**] **enable**

Example:

```
switch(config)# enable
```

The no form of this command disables Smart Call Home.

Configure Source Interface

Use this procedure to identify the source-interface in a VRF through which the packet is sent.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
                        switch(config-callhome)#
```

Step 3 Define the source interface for call home http(s) transfer, and when call home is used as the transport for Smart Licensing using the command **source interface** *interface_name*

Example:

```
switch(config-callhome)# source-interface Ethernet1/1
```

Test Smart Call Home Configuration

You can generate a test message to test your Smart Call Home communications.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Enter Smart Call Home configuration mode using the command **callhome**

Example:

```
switch(config)# callhome
  switch(config-callhome)#
```

Step 3 Send the specified Smart Call Home test message to all configured destinations using the command **callhome send** [**configuration** | **diagnostic**]

Example:

```
switch(config-callhome)# callhome send diagnostic
```

Step 4 Send a test message to all configured destinations using the command **callhome test**

Example:

```
switch(config-callhome)# callhome test
```

Step 5 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Additional Reference Information

Table 13: The event triggers and their Smart Call Home message severity levels

Alert Group	Event Name	Description	Smart Call Home Severity Level
Configuration	PERIODIC_CONFIGURATION	Periodic configuration update message.	2
Diagnostic	DIAGNOSTIC_MAJOR_ALERT	GOLD generated a major alert.	7
Diagnostic	DIAGNOSTIC_MINOR_ALERT	GOLD generated a minor alert.	4
Diagnostic	DIAGNOSTIC_NORMAL_ALERT	Smart Call Home generated a normal diagnostic alert.	2
Environmental and CISCO_TAC	FAN_FAILURE	Cooling fan has failed.	5
Environmental and CISCO_TAC	POWER_SUPPLY_ALERT	Power supply warning has occurred.	6
Environmental and CISCO_TAC	POWER_SUPPLY_FAILURE	Power supply has failed.	6

Alert Group	Event Name	Description	Smart Call Home Severity Level
Environmental and CISCO_TAC	POWER_SUPPLY_SHUTDOWN	Power supply has shut down.	6
Environmental and CISCO_TAC	TEMPERATURE_ALARM	Thermal sensor going bad.	6
Environmental and CISCO_TAC	TEMPERATURE_MAJOR_ALARM	Thermal sensor indicates temperature has reached operating major threshold.	6
Environmental and CISCO_TAC	TEMPERATURE_MINOR_ALARM	Thermal sensor indicates temperature has reached operating minor threshold.	4
Inventory and CISCO_TAC	COLD_BOOT	Switch is powered up and reset to a cold boot sequence.	2
Inventory and CISCO_TAC	HARDWARE_INSERTION	New piece of hardware has been inserted into the chassis.	2
Inventory and CISCO_TAC	HARDWARE_REMOVAL	Hardware has been removed from the chassis.	2
Inventory and CISCO_TAC	PERIODIC_INVENTORY	Periodic inventory message has been generated.	2
License	LICENSE_VIOLATION	Feature in use is not licensed and is turned off after grace period expiration.	6
Line module Hardware and CISCO_TAC	LINEmodule_FAILURE	Module operation has failed.	7
Supervisor Hardware and CISCO_TAC	SUP_FAILURE	Supervisor module operation has failed.	7
Syslog-group-port	PORT_FAILURE	syslog message that corresponds to the port facility has been generated.	6

Alert Group	Event Name	Description	Smart Call Home Severity Level
Syslog-group-port	SYSLOG_ALERT	syslog alert message has been generated. **Note** Link up/down syslog messages do not trigger Smart Call Home messages or alert notifications.	5
System and CISCO_TAC	SW_CRASH	Software process has failed with a stateless restart, indicating an interruption of a service. Messages are sent for process crashes on supervisor modules.	5
System and CISCO_TAC	SW_SYSTEM_INCONSISTENT	Inconsistency has been detected in software or file system.	5
Test and CISCO_TAC	TEST	User generated test has occurred.	2

Smart Call Home supports the following message formats:

- Short Text Message Format
- Common Event Message Fields
- Alert Group Message Fields
- Reactive and Proactive Event Message Fields
- Inventory Event Message Fields
- User-Generated Test Message Fields

Table 14: Short text formatting option for all message types

Data Item	Description
Device identification	Configured device name
Date/time stamp	Time stamp of the triggering event
Error isolation message	Plain English description of triggering event
Alarm urgency level	Error level such as that applied to system message

Table 15: First set of common event message fields for full text or XML messages

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Timestamp	Date and time stamp of event in ISO time notation: YYYY-MM-DD HH:MM:SS GMT+HH:MM.	/aml/header/time
Message name	Name of message.	/aml/header/name
Message type	Name of message type, such as reactive or proactive.	/aml/header/type
Message group	Name of alert group, such as syslog.	/aml/header/group
Severity level	Severity level of message.	/aml/header/level
Source ID	Product type for routing, such as the Cisco Nexus 9000 Series switch.	/aml/header/source
Device ID	Unique device identifier (UDI) for the end device that generated the message. This field should be empty if the message is nonspecific to a device. The format is **type@Sid@serial**. • *type* is the product model number from the backplane IDPROM. • @ is a separator character. • *Sid* is C, identifying the serial ID as a chassis serial number. • *serial* is the number identified by the Sid field. An example is N9K-C9508@C@12345678.	/aml/ header/deviceId
Customer ID	Optional user-configurable field used for contract information or other ID by any support service.	/aml/ header/customerID
Contract ID	Optional user-configurable field used for contract information or other ID by any support service.	/aml/ header /contractId
Site ID	Optional user-configurable field used for Cisco-supplied site ID or other data meaningful to alternate support service.	/aml/ header/siteId

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Server ID	If the message is generated from the device, this ID is the unique device identifier (UDI) of the device. The format is **type@Sid@serial**. • *type* is the product model number from the backplane IDPROM. • @ is a separator character. • *Sid* is C, identifying the serial ID as a chassis serial number. • *serial* is the number identified by the Sid field. An example is N9K-C9508@C@12345678.	/aml/header/serverId
Message description	Short text that describes the error.	/aml/body/msgDesc
Device name	Node that experienced the event (hostname of the device).	/aml/body/sysName
Contact name	Name of person to contact for issues associated with the node that experienced the event.	/aml/body/sysContact
Contact email	Email address of person identified as the contact for this unit.	/aml/body/sysContactEmail
Contact phone number	Phone number of the person identified as the contact for this unit.	/aml/body/sysContactPhone Number
Street address	Optional field that contains the street address for RMA part shipments associated with this unit.	/aml/body/sysStreetAddress
Model name	Model name of the device (the specific model as part of a product family name).	/aml/body/chassis/name
Serial number	Chassis serial number of the unit.	/aml/body/chassis/serialNo
Chassis part number	Top assembly number of the chassis.	/aml/body/chassis/partNo

Table 16: Fields specific to alert group messages for full text and XML

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Command output name	Exact name of the issued CLI command.	/aml/attachments/attachment/name

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Attachment type	Specific command output.	/aml/attachments/attachment/type
MIME type	Either plain text or encoding type.	/aml/attachments/attachment/mime
Command output text	Output of command automatically executed.	/aml/attachments/attachment/atdata



Note These fields may be repeated if multiple CLI commands are executed for an alert group.

Table 17: Reactive and Proactive event message format for full text or XML messages

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Chassis hardware version	Hardware version of chassis.	/aml/body/chassis/hwVersion
Supervisor module software version	Top-level software version.	/aml/body/chassis/swVersion
Affected FRU name	Name of the affected FRU that is generating the event message.	/aml/body/fru/name
Affected FRU serial number	Serial number of the affected FRU.	/aml/body/fru/serialNo
Affected FRU part number	Part number of the affected FRU.	/aml/body/fru/partNo
FRU slot	Slot number of the FRU that is generating the event message.	/aml/body/fru/slot
FRU hardware version	Hardware version of the affected FRU.	/aml/body/fru/hwVersion
FRU software version	Software version(s) that is running on the affected FRU.	/aml/body/fru/swVersion

Table 18: Inventory event message format for full text or XML messages

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Chassis hardware version	Hardware version of the chassis.	/aml/body/chassis/hwVersion
Supervisor module software version	Top-level software version.	/aml/body/chassis/swVersion
FRU name	Name of the affected FRU that is generating the event message.	/aml/body/fru/name
FRU s/n	Serial number of the FRU.	/aml/body/fru/serialNo
FRU part number	Part number of the FRU.	/aml/body/fru/partNo
FRU slot	Slot number of the FRU.	/aml/body/fru/slot
FRU hardware version	Hardware version of the FRU.	/aml/body/fru/hwVersion

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
FRU software version	Software version(s) that is running on the FRU.	/aml/body/fru/swVersion

Table 19: User-generated test message format for full text or XML

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Process ID	Unique process ID.	/aml/body/process/id
Process state	State of process (for example, running or halted).	/aml/body/process/processState
Process exception	Exception or reason code.	/aml/body/process/exception

Sample Syslog Alert Notification

- This sample shows the full-text format for a syslog port alert-group notification

```
Severity Level:5
Series:Nexus9000
Switch Priority:0
Device Id:N9K-C9508@C@TXX12345678
Server Id:N9K-C9508C@TXX12345678
Time of Event:2013-05-17 16:31:33 GMT+0000 Message Name:
Message Type:syslog
System Name:dc3-test
Contact Name:Jay Tester
Contact Email:contact@example.com
Contact Phone:+91-80-1234-5678
Street Address:#1 Any Street
Event Description:SYSLOG_ALERT 2013 May 17 16:31:33 dc3-test %ETHPORT-2-IF_SEQ_ERROR:
Error (0x20) while communicating with component MTS_SAP_ELTM
opcode:MTS_OPC_ETHPM_PORT_PHY_CLEANUP (for:RID_PORT: Ethernet3/1)

syslog_facility:ETHPORT
start chassis information:
Affected Chassis:N9K-C9508
Affected Chassis Serial Number:TXX12345678 Affected Chassis Hardware Version:0.405
Affected Chassis Software Version:6.1(2) Affected Chassis Part No:11-11111-11 end chassis
 information:
start attachment
   name:show logging logfile | tail -n 200
   type:text
   data:
   2013 May 17 10:57:51 dc3-test %SYSLOG-1-SYSTEM_MSG : Logging logfile (messages)
cleared by user
   2013 May 17 10:57:53 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
 /dev/ttyS0 /dev/ttyS0_console
   2013 May 17 10:58:35 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
 /dev/ttyS0 /dev/ttyS0_console
   2013 May 17 10:59:00 dc3-test %DAEMON-3-SYSTEM_MSG: error: setsockopt IP_TOS 16:
Invalid argument: - sshd[14484]
   2013 May 17 10:59:05 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
 /dev/ttyS0 /dev/ttyS0_console
   2013 May 17 12:11:18 dc3-test %SYSMGR-STANDBY-5-SUBPROC_TERMINATED: "System Manager
 (gsync controller)" (PID 12000) has finished with error code
SYSMGR_EXITCODE_GSYNCFAILED_NONFATAL (12).
   2013 May 17 16:28:03 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
 /dev/ttyS0 /dev/ttyS0_console
```

```
  2013 May 17 16:28:44 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2579 with message
 Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
  2013 May 17 16:28:44 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 3504)
hasn't caught signal 9 (no core).
  2013 May 17 16:29:08 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2579 with message
 Core not generated by system for eltm(0). WCOREDUMP(9) returned zero.
  2013 May 17 16:29:08 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 23210)
 hasn't caught signal 9 (no core).
  2013 May 17 16:29:17 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2579 with message
 Core not generated by system for eltm(0). WCOREDUMP(9) returned zero.
  2013 May 17 16:29:17 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 23294)
 hasn't caught signal 9 (no core).
  2013 May 17 16:29:25 dc3-test %SYSMGR-2-HASWITCHOVER_PRE_START: This supervisor is
becoming active (pre-start phase).
  2013 May 17 16:29:25 dc3-test %SYSMGR-2-HASWITCHOVER_START: This supervisor is
becoming active.
  2013 May 17 16:29:26 dc3-test %USER-3-SYSTEM_MSG: crdcfg_get_srvinfo: mts_send failed
 - device_test
  2013 May 17 16:29:27 dc3-test %NETSTACK-3-IP_UNK_MSG_MAJOR: netstack [4336]
Unrecognized message from MRIB. Major type 1807
  2013 May 17 16:29:27 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is DOWN
  2013 May 17 16:29:28 dc3-test %SYSMGR-2-SWITCHOVER_OVER: Switchover completed.
  2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 2 - ntpd[19045]

  2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 10 -
ntpd[19045]
  2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:ipv6 only defined - ntpd[19045]

  2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:bindv6 only defined -
ntpd[19045]
  2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 2 - ntpd[19045]

  2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 0 - ntpd[19045]

  2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 0 - ntpd[19045]

  2013 May 17 16:29:28 dc3-test %NETSTACK-3-CLIENT_GET: netstack [4336] HA client
filter recovery failed (0)
  2013 May 17 16:29:28 dc3-test %NETSTACK-3-CLIENT_GET: netstack [4336] HA client
filter recovery failed (0)
  2013 May 17 16:29:29 dc3-test %DAEMON-3-SYSTEM_MSG: ssh disabled, removing -
dcos-xinetd[19072]
  2013 May 17 16:29:29 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19072]
  2013 May 17 16:29:31 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19073]
  2013 May 17 16:29:32 dc3-test %DAEMON-3-SYSTEM_MSG: ssh disabled, removing -
dcos-xinetd[19079]
  2013 May 17 16:29:32 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19079]
  2013 May 17 16:29:34 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is UP
  2013 May 17 16:29:34 dc3-test %DAEMON-3-SYSTEM_MSG: ssh disabled, removing -
dcos-xinetd[19105]
  2013 May 17 16:29:34 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19105]
  2013 May 17 16:29:35 dc3-test %PLATFORM-2-PS_AC_IN_MISSING: Power supply 2 present
but all AC inputs are not connected, ac-redundancy might be affected
  2013 May 17 16:29:35 dc3-test %PLATFORM-2-PS_AC_IN_MISSING: Power supply 3 present
but all AC inputs are not connected, ac-redundancy might be affected
  2013 May 17 16:29:38 dc3-test %CALLHOME-2-EVENT: SUP_FAILURE
  2013 May 17 16:29:46 dc3-test vsh[19166]: CLIC-3-FAILED_EXEC: Can not exec command
<more> return code <14>
  2013 May 17 16:30:24 dc3-test vsh[23810]: CLIC-3-FAILED_EXEC: Can not exec command
<more> return code <14>
```

```
   2013 May 17 16:30:24 dc3-test vsh[23803]: CLIC-3-FAILED_EXEC: Can not exec command
<more> return code <14>
   2013 May 17 16:30:24 dc3-test vsh[23818]: CLIC-3-FAILED_EXEC: Can not exec command
<more> return code <14>
   2013 May 17 16:30:47 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message
 Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
   2013 May 17 16:30:47 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 4820)
hasn't caught signal 9 (no core).
   2013 May 17 16:31:02 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message
 Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
   2013 May 17 16:31:02 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 24239)
 hasn't caught signal 9 (no core).
   2013 May 17 16:31:14 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message
 Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
   2013 May 17 16:31:14 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 24401)
 hasn't caught signal 9 (no core).
   2013 May 17 16:31:23 dc3-test %CALLHOME-2-EVENT: SW_CRASH alert for service: eltm
   2013 May 17 16:31:23 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message
 Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
   2013 May 17 16:31:23 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service "eltm" (PID 24407)
 hasn't caught signal 9 (no core).
   2013 May 17 16:31:24 dc3-test vsh[24532]: CLIC-3-FAILED_EXEC: Can not exec command
<more> return code <14>
   2013 May 17 16:31:24 dc3-test vsh[24548]: CLIC-3-FAILED_EXEC: Can not exec command
<more> return code <14>
   2013 May 17 16:31:24 dc3-test vsh[24535]: CLIC-3-FAILED_EXEC: Can not exec command
<more> return code <14>
   2013 May 17 16:31:33 dc3-test %NETSTACK-3-INTERNAL_ERROR: netstack [4336] (null)
   2013 May 17 16:31:33 dc3-test %ETHPORT-2-IF_SEQ_ERROR: Error (0x20) while communicating
 with component MTS_SAP_ELTM opcode:MTS_OPC_ETHPM_PORT_PHY_CLEANUP (for:RID_PORT:
Ethernet3/1) end attachment start attachment
   type:text
   data:

   dc3-test interfaces:
        Ethernet3/1    Ethernet3/2    Ethernet3/3
        Ethernet3/4    Ethernet3/5    Ethernet3/6
        Ethernet3/7    Ethernet3/8    Ethernet3/9
        Ethernet3/10   Ethernet3/11   Ethernet3/12
        Ethernet3/13   Ethernet3/14   Ethernet3/15
        Ethernet3/16   Ethernet3/17   Ethernet3/18
        Ethernet3/19   Ethernet3/20   Ethernet3/21
        Ethernet3/22   Ethernet3/23   Ethernet3/24
        Ethernet3/25   Ethernet3/29   Ethernet3/30
        Ethernet3/31   Ethernet3/32   Ethernet3/33
        Ethernet3/34   Ethernet3/35   Ethernet3/36
        Ethernet3/37   Ethernet3/38   Ethernet3/39
        Ethernet3/40   Ethernet3/41   Ethernet3/42
        Ethernet3/43   Ethernet3/44   Ethernet3/45
        Ethernet3/46   Ethernet3/47   Ethernet3/48
end attachment
start attachment
   type:text
   data:
end attachment
start attachment
   name:show license usage
   type:text
   data:
   Feature Ins Lic Status Expiry Date Comments
              Count
   --------------------------------------------------------------------------------
   LAN_ENTERPRISE_SERVICES_PKG Yes - Unused Never -
   --------------------------------------------------------------------------------
```

```
end attachment
```

- This sample shows the XML format for a syslog port alert-group notification

```
<?xml version="1.0" encoding="UTF-8" ?>
<soap-env:Envelope xmlns:soap-env="http://www.w3.org/2003/05/soap-envelope">
<soap-env:Header>
<aml-session:Session xmlns:aml-session="http://www.cisco.com/2004/01/aml-session"
soap-env:mustUnderstand="true"
soap-env:role="http://www.w3.org/2003/05/soap-envelope/role/next">
<aml-session:To>http://tools.cisco.com/neddce/services/DDCEService</aml-session:To>
<aml-session:Path>
<aml-session:Via>http://www.cisco.com/appliance/uri</aml-session:Via>
</aml-session:Path>
<aml-session:From>http://www.cisco.com/appliance/uri</aml-session:From>
<aml-session:MessageId>1004:TXX12345678:478F82E6</aml-session:MessageId>
</aml-session:Session>
</soap-env:Header>
<soap-env:Body>
<aml-block:Block xmlns:aml-block="http://www.cisco.com/2004/01/aml-block">
<aml-block:Header>
<aml-block:Type>http://www.cisco.com/2005/05/callhome/syslog</aml-block:Type>
<aml-block:CreationDate>2013-05-17 16:31:33 GMT+0000</aml-block:CreationDate>
<aml-block:Builder> <aml-block:Name>DC3</aml-block:Name>
<aml-block:Version>4.1</aml-block:Version>
</aml-block:Builder>
<aml-block:BlockGroup>
<aml-block:GroupId>1005:TXX12345678:478F82E6</aml-block:GroupId>
<aml-block:Number>0</aml-block:Number>
<aml-block:IsLast>true</aml-block:IsLast>
<aml-block:IsPrimary>true</aml-block:IsPrimary>
<aml-block:WaitForPrimary>false</aml-block:WaitForPrimary>
</aml-block:BlockGroup>
<aml-block:Severity>5</aml-block:Severity>
</aml-block:Header>
<aml-block:Content>
<ch:CallHome xmlns:ch="http://www.cisco.com/2005/05/callhome" version="1.0">
<ch:EventTime>2013-05-17 16:31:33 GMT+0000</ch:EventTime>
<ch:MessageDescription>SYSLOG_ALERT 2013 May 17 16:31:33 dc3-test %ETHPORT-2-IF_SEQ_ERROR:
 Error (0x20) while communicating with component MTS_SAP_ELTM
opcode:MTS_OPC_ETHPM_PORT_PHY_CLEANUP (for:RID_PORT: Ethernet3/1) </ch:MessageDescription>
<ch:Event> <ch:Type>syslog</ch:Type> <ch:SubType></ch:SubType> <ch:Brand>Cisco</ch:Brand>
 <ch:Series>Nexus9000</ch:Series> </ch:Event> <ch:CustomerData> <ch:UserData>
<ch:Email>contact@example.com</ch:Email>
</ch:UserData>
<ch:ContractData>
<ch:DeviceId>N9K-C9508@C@TXX12345678</ch:DeviceId>
</ch:ContractData>
<ch:SystemInfo>
<ch:Name>dc3-test</ch:Name>
<ch:Contact>Jay Tester</ch:Contact> <ch:ContactEmail>contact@example.com</ch:ContactEmail>
<ch:ContactPhoneNumber>+91-80-1234-5678</ch:ContactPhoneNumber>
<ch:StreetAddress>#1, Any Street</ch:StreetAddress> </ch:SystemInfo> </ch:CustomerData>
 <ch:Device> <rme:Chassis xmlns:rme="http://www.cisco.com/rme/4.1">
<rme:Model>N9K-C9508</rme:Model>
<rme:HardwareVersion>0.405</rme:HardwareVersion>
<rme:SerialNumber>TXX12345678</rme:SerialNumber>
</rme:Chassis>
</ch:Device>
</ch:CallHome>
</aml-block:Content>
<aml-block:Attachments>
<aml-block:Attachment type="inline">
<aml-block:Name>show logging logfile | tail -n 200</aml-block:Name> <aml-block:Data
```

```
encoding="plain">
<![CDATA[2013 May 17 10:57:51 dc3-test %SYSLOG-1-SYSTEM_MSG : Logging logfile (messages)
 cleared by user
2013 May 17 10:57:53 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
/dev/ttyS0 /dev/ttyS0_console
2013 May 17 10:58:35 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
/dev/ttyS0 /dev/ttyS0_console
2013 May 17 10:59:00 dc3-test %DAEMON-3-SYSTEM_MSG: error: setsockopt IP_TOS 16: Invalid
 argument: - sshd[14484]
2013 May 17 10:59:05 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
/dev/ttyS0 /dev/ttyS0_console
2013 May 17 12:11:18 dc3-test %SYSMGR-STANDBY-5-SUBPROC_TERMINATED: \"System Manager
(gsync controller)\" (PID 12000) has finished with error code
SYSMGR_EXITCODE_GSYNCFAILED_NONFATAL (12).
2013 May 17 16:28:03 dc3-test %VSHD-5-VSHD_SYSLOG_CONFIG_I: Configuring console from
/dev/ttyS0 /dev/ttyS0_console
2013 May 17 16:28:44 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2579 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2013 May 17 16:28:44 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 3504)
hasn&apos;t caught signal 9 (no core).
2013 May 17 16:29:08 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2579 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2013 May 17 16:29:08 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 23210)
hasn&apos;t caught signal 9 (no core).
2013 May 17 16:29:17 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2579 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2013 May 17 16:29:17 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 23294)
hasn&apos;t caught signal 9 (no core).
2013 May 17 16:29:25 dc3-test %SYSMGR-2-HASWITCHOVER_PRE_START: This supervisor is
becoming active (pre-start phase).
2013 May 17 16:29:25 dc3-test %SYSMGR-2-HASWITCHOVER_START: This supervisor is becoming
 active.
2013 May 17 16:29:26 dc3-test %USER-3-SYSTEM_MSG: crdcfg_get_srvinfo: mts_send failed
- device_test
2013 May 17 16:29:27 dc3-test %NETSTACK-3-IP_UNK_MSG_MAJOR: netstack [4336] Unrecognized
 message from MRIB. Major type 1807
2013 May 17 16:29:27 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is DOWN
2013 May 17 16:29:28 dc3-test %SYSMGR-2-SWITCHOVER_OVER: Switchover completed.
2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 2 - ntpd[19045]
2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 10 - ntpd[19045]
2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:ipv6 only defined - ntpd[19045]
2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:bindv6 only defined - ntpd[19045]
2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 2 - ntpd[19045]
2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 0 - ntpd[19045]
2013 May 17 16:29:28 dc3-test %DAEMON-3-SYSTEM_MSG: ntp:socket family : 0 - ntpd[19045]
2013 May 17 16:29:28 dc3-test %NETSTACK-3-CLIENT_GET: netstack [4336] HA client filter
 recovery failed (0)
2013 May 17 16:29:28 dc3-test %NETSTACK-3-CLIENT_GET: netstack [4336] HA client filter
 recovery failed (0)
2013 May 17 16:29:29 dc3-test %DAEMON-3-SYSTEM_MSG: ssh disabled, removing -
dcos-xinetd[19072]
2013 May 17 16:29:29 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19072]
2013 May 17 16:29:31 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19073]
2013 May 17 16:29:32 dc3-test %DAEMON-3-SYSTEM_MSG: ssh disabled, removing -
dcos-xinetd[19079]
2013 May 17 16:29:32 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19079]
2013 May 17 16:29:34 dc3-test %IM-5-IM_INTF_STATE: mgmt0 is UP
2013 May 17 16:29:34 dc3-test %DAEMON-3-SYSTEM_MSG: ssh disabled, removing -
dcos-xinetd[19105]
2013 May 17 16:29:34 dc3-test %DAEMON-3-SYSTEM_MSG: Telnet disabled, removing -
dcos-xinetd[19105]
```

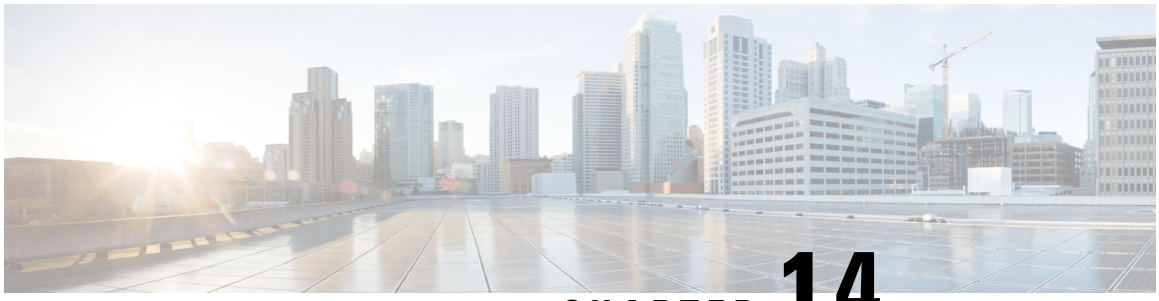
```
2013 May 17 16:29:35 dc3-test %PLATFORM-2-PS_AC_IN_MISSING: Power supply 2 present but
 all AC inputs are not connected, ac-redundancy might be affected
2013 May 17 16:29:35 dc3-test %PLATFORM-2-PS_AC_IN_MISSING: Power supply 3 present but
 all AC inputs are not connected, ac-redundancy might be affected
2013 May 17 16:29:38 dc3-test %CALLHOME-2-EVENT: SUP_FAILURE
2013 May 17 16:29:46 dc3-test vsh[19166]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2013 May 17 16:30:24 dc3-test vsh[23810]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2013 May 17 16:30:24 dc3-test vsh[23803]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2013 May 17 16:30:24 dc3-test vsh[23818]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2013 May 17 16:30:47 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2013 May 17 16:30:47 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 4820)
hasn&apos;t caught signal 9 (no core).
2013 May 17 16:31:02 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2013 May 17 16:31:02 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 24239)
hasn&apos;t caught signal 9 (no core).
2013 May 17 16:31:14 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2013 May 17 16:31:14 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 24401)
hasn&apos;t caught signal 9 (no core).
2013 May 17 16:31:23 dc3-test %CALLHOME-2-EVENT: SW_CRASH alert for service: eltm
2013 May 17 16:31:23 dc3-test %SYSMGR-3-BASIC_TRACE: core_copy: PID 2630 with message
Core not generated by system for eltm(0). WCOREDUMP(9) returned zero .
2013 May 17 16:31:23 dc3-test %SYSMGR-2-SERVICE_CRASHED: Service \"eltm\" (PID 24407)
hasn&apos;t caught signal 9 (no core).
2013 May 17 16:31:24 dc3-test vsh[24532]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2013 May 17 16:31:24 dc3-test vsh[24548]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2013 May 17 16:31:24 dc3-test vsh[24535]: CLIC-3-FAILED_EXEC: Can not exec command
&lt;more&gt; return code &lt;14&gt;
2013 May 17 16:31:33 dc3-test %NETSTACK-3-INTERNAL_ERROR: netstack [4336] (null)
2013 May 17 16:31:33 dc3-test %ETHPORT-2-IF_SEQ_ERROR: Error (0x20) while communicating
 with component MTS_SAP_ELTM opcode:MTS_OPC_ETHPM_PORT_PHY_CLEANUP (for:RID_PORT:
Ethernet3/1) ]]> </aml-block:Data> </aml-block:Attachment> <aml-block:Attachment
type="inline"> <aml-block:Name> <aml-block:Data encoding="plain"> <![CDATA[
dc3-test interfaces:
   Ethernet3/1     Ethernet3/2     Ethernet3/3
   Ethernet3/4     Ethernet3/5     Ethernet3/6
   Ethernet3/7     Ethernet3/8     Ethernet3/9
   Ethernet3/10    Ethernet3/11    Ethernet3/12
   Ethernet3/13    Ethernet3/14    Ethernet3/15
   Ethernet3/16    Ethernet3/17    Ethernet3/18
   Ethernet3/19    Ethernet3/20    Ethernet3/21
   Ethernet3/22    Ethernet3/23    Ethernet3/24
   Ethernet3/25    Ethernet3/26    Ethernet3/27
   Ethernet3/28    Ethernet3/29    Ethernet3/30
   Ethernet3/31    Ethernet3/32    Ethernet3/33
   Ethernet3/34    Ethernet3/35    Ethernet3/36
   Ethernet3/37    Ethernet3/38    Ethernet3/39
   Ethernet3/40    Ethernet3/41    Ethernet3/42
   Ethernet3/43    Ethernet3/44    Ethernet3/45
   Ethernet3/46    Ethernet3/47    Ethernet3/48


]]>
</aml-block:Data>
</aml-block:Attachment>
<aml-block:Attachment type="inline">
```

```
<aml-block:Name> <aml-block:Data encoding="plain"> <!> </aml-block:Data>
</aml-block:Attachment> <aml-block:Attachment type="inline"> <aml-block:Name>show license
 usage</aml-block:Name> <aml-block:Data encoding="plain">
<![CDATA[Feature Ins Lic Status Expiry Date Comments
                   Count
--------------------------------------------------------------------------------
LAN_ENTERPRISE_SERVICES_PKG Yes - Unused Never -
--------------------------------------------------------------------------------
]]>
</aml-block:Data>
</aml-block:Attachment>
</aml-block:Attachments>
</aml-block:Block>
</soap-env:Body>
</soap-env:Envelope>
```

MIBs

MIBs related to Smart Call Home - To locate and download supported MIBs, go to the following URL: https://cisco.github.io/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html



CHAPTER 14

Session Manager

This chapter describes how to configure Session Manager on Cisco NX-OS devices.

This chapter contains the following sections:

- Session Manager, on page 225
- Prerequisites for Session Manager, on page 226
- Guidelines and limitations for Session Manager, on page 226
- Configure Session Manager, on page 226
- Commands for verifying Session Manager configuration, on page 229
- Configuration example for Session Manager, on page 229
- Additional references, on page 230

Session Manager

Session Manager allows you to implement your configuration changes in batch mode.

Session Manager works in five phases:

- Configuration session—Creates a list of commands that you want to implement in Session Manager mode.
- Validation—Provides a basic semantic check on your configuration. NX-OS returns an error if the semantic check fails on any part of the configuration.
- Verification—Verifies the configuration as a whole, based on the existing hardware and software configuration and resources. NX-OS returns an error if the configuration does not pass this verification phase.
- Commit—NX-OS verifies the complete configuration and applies the changes to the device. If a failure occurs, NX-OS reverts to the original configuration.
- Abort—Discards the configuration changes before implementation.

You can optionally end a configuration session without committing the changes. You can also save a configuration session.

High Availability

Session Manager sessions remain available after a supervisor switchover. Sessions are not persistent across a software reload.

Prerequisites for Session Manager

Make sure that you have the privilege level required to support the Session Manager commands that you plan to use.

Guidelines and limitations for Session Manager

The configuration guidelines and limitations for Session Manager are:

- Configuration for only one service access point (SAP) can be performed using one session.
- Configuration sessions are not persistent across reloads.
- Session Manager supports only access control list (ACL) and quality of service (QoS) features.
- You can create up to 32 configuration sessions.
- You can configure a maximum of 20,000 commands across all sessions.
- You cannot simultaneously execute configuration commands in more than one configuration session or configuration terminal mode. Parallel configurations (for example, one configuration session and one configuration terminal) could cause validation or verification failures in the configuration session.
- If an interface reloads while you are configuring it in a configuration session, Session Manager can accept the commands even if the interface is not present in the device.
- Beginning from the mentioned NX-OS Release, any network admin user can configure or delete existing sessions.
    - 10.5(1)F

Configure Session Manager

Note Be aware that the NX-OS commands might differ from IOS commands.

Create a session

You can create up to 32 configuration sessions.

Procedure

Step 1 Display the contents of the session using the **configure session** *name* command.

Example:

```
switch(config-s)# save bootflash:sessions/myACLs
```

This command creates a configuration session and enters session configuration mode. The name can be any alphanumeric string.

Step 2 (Optional) Display the contents of the session using the **show configuration session** [*name*] command.

Example:

```
switch(config-s)# show configuration session myACLs
```

Step 3 (Optional) Save the session to a file using the **save** *location* command. The location can be in `bootflash:`, `slot0:`, or `volatile:`.

Example:

```
switch(config-s)# save bootflash:sessions/myACLs
```

Configure ACLs in a session

You can configure ACLs within a configuration session.

Procedure

Step 1 Create a configuration session and enter session configuration mode using the **configure session** *name* command. The name can be any alphanumeric string.

Example:

```
switch# configure session myacls
switch(config-s)#
```

Step 2 Create an ACL and enter a configuration mode for that ACL using the **ip access-list** *name* command.

Example:

```
switch(config-s)# ip access-list acl1
switch(config-s-acl)#
```

Step 3 (Optional) Add a permit statement to the ACL using the **permit** *protocol source destination* command.

Example:

```
switch(config-s-acl)# permit tcp any any
```

Step 4 Enter interface configuration mode using the **interface** *interface-type number* command.

Example:

```
switch(config-s-acl)# interface ethernet 2/1
switch(config-s-if)#
```

Step 5 Specify the direction of traffic the access group is applied to using the **ip access-group** *name* {**in** | **out**} command.

Example:

```
switch(config-s-if)# ip access-group acl1 in
```

Step 6 (Optional) Display the contents of the session using the **show configuration session** [*name*] command.

Example:

```
switch(config-s-if)# show configuration session myacls
```

Verify a session

Use this command in session mode to verify a session.

Procedure

Verify the configuration as a whole, based on the existing hardware and software configuration and resources using the **verify** [**verbose**] command.

Example:

```
switch(config-s)# verify
```

NX-OS returns an error if the configuration does not pass this verification.

Commit a session

Use this command in session mode to commit a session.

Procedure

Validate the configuration changes made in the current session and apply valid changes to the device using the **commit** [**verbose**] command.

Example:

```
switch(config-s)# commit
```

If the validation fails, NX-OS reverts to the original configuration.

Save a session

Use this command in session mode to save a session.

Procedure

(Optional) Save the session to a file using the **save** *location* command. The location can be in bootflash:, slot0:, or volatile:.

Example:

```
switch(config-s)# save bootflash:sessions/myACLs
```

Discard a Session

Use this command in session mode to discard a session.

Procedure

Discard the configuration session without applying the changes using the **abort** command.

Example:

```
switch(config-s)# abort
switch#
```

Commands for verifying Session Manager configuration

To display the Session Manager configuration information, perform any of the required show commands listed in the table.

Command	Purpose
show configuration session [*name*]	Displays the contents of the configuration session.
show configuration session status [*name*]	Displays the status of the configuration session.
show configuration session summary	Displays a summary of all the configuration sessions.

Configuration example for Session Manager

This example shows how to create and commit an ACL configuration using Session Manager.

```
switch# configure session ACL_tcp_in
Config Session started, Session ID is 1
```

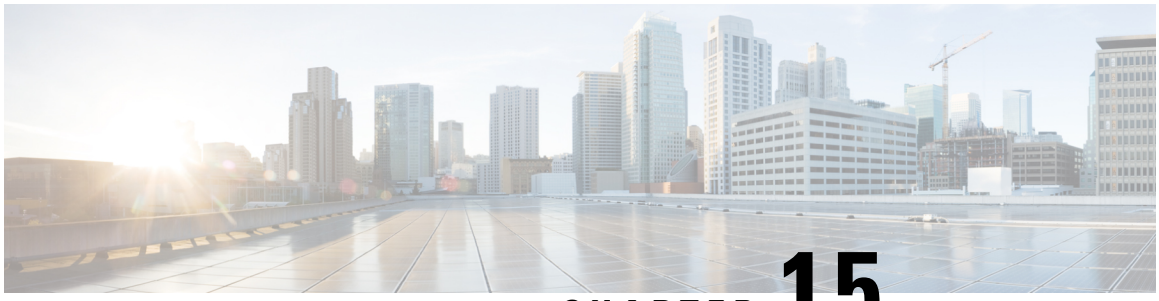
```
Enter configuration commands, one per line. End with CNTL/Z.
switch(config-s)# ip access-list ACL1
switch(config-s-acl)# permit tcp any any
switch(config)# interface e 7/1
switch(config-if)# ip access-group ACL1 in
switch(config-if)# exit
switch(config)# exit
switch# config session ACL_tcp_in
Config Session started, Session ID is 1
Enter configuration commands, one per line. End with CNTL/Z.
switch(config-s)# verify
Verification Successful
switch(config-s)# commit
Commit Successful
switch#
```

Additional references

Related documents

This reference provides information about related documents for configuration files.

Related Topic	Document Title
Configuration files	*Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide*



CHAPTER 15

Scheduler

- Overview, on page 231
- Guidelines and Limitations, on page 232
- Enable or Disable the Scheduler, on page 233
- Define Scheduler Log File Size, on page 234
- Configure Remote User Authentication, on page 235
- Define Job, on page 236
- Delete Job, on page 237
- Define Timetable, on page 237
- Clear Scheduler Log File, on page 239
- Verify Scheduler Configuration, on page 240
- Configuration Examples, on page 240

Overview

The scheduler allows you to define and set a timetable for maintenance activities such as the following:

- Quality of service (QoS) policy changes
- Data backup
- Saving a configuration

Jobs

Jobs consist of a single command or multiple commands that define routine activities. Jobs can be scheduled one time or at periodic intervals. The scheduler defines a job and its timetable as follows:

- Job—A routine task or tasks defined as a command list and completed according to a specified schedule.
- Schedule—The timetable for completing a job. You can assign multiple jobs to a schedule. A schedule is defined as either periodic or one-time only:
  - Periodic mode—A recurring interval that continues until you delete the job. You can configure the following types of intervals:
    - Daily—A job is completed once a day.
    - Weekly—A job is completed once a week.

- Monthly—A job is completed once a month.
- Delta—A job begins at the specified start time and then at specified intervals (days:hours:minutes).

- One-time mode—A job is completed only once at a specified time.

Features

- Logs - The scheduler maintains a log file containing the job output. If the size of the job output is greater than the size of the log file, the output is truncated.
- Remote User Authentication - Before starting a job, the scheduler authenticates the user who created the job. Since user credentials from a remote authentication are not retained long enough to support a scheduled job, you need to locally configure the authentication passwords for users who create jobs. These passwords are part of the scheduler configuration and are not considered a locally configured user.

  Before starting the job, the scheduler validates the local password against the password from the remote authentication server.
- High Availability - Scheduled jobs remain available after a supervisor switchover or a software reload.
- Default Settings -

  This table lists the scheduler default settings.

Parameters	Default
Scheduler state	Disabled
Log file size	16 KB

Guidelines and Limitations

The scheduler has the following configuration guidelines and limitations:

Prerequisites

- You must enable any conditional features before you can configure those features in a job.
- You must have a valid license installed for any licensed features that you want to configure in the job.
- You must have network-admin user privileges to configure a scheduled job.

General Guidelines

- The scheduler can fail if it encounters one of the following while performing a job:
  - Verify that you have configured the time. The scheduler does not apply a default timetable. If you create a schedule and assign jobs and do not configure the time, the job is not started.
  - While defining a job, verify that no interactive or disruptive commands (for example, **copy bootflash:** *file* **ftp:** *URI* , **write erase** , **reload** , and other similar commands) are specified because the job

is started and conducted noninteractively. When a reload job is scheduled for a given time and executed, the switch goes into a boot loop. Hence it should not be used in scheduler configuration.

- The scheduler accepts `start_time` in the past for any schedule with a repeat option in the **time** command under the schedule mode configuration. It then throws a warning that the entered start time is in the past. The `start_time` of any schedule will always remain the same as it was in the beginning, across reboot, and even after reapplying the previous saved configuration.
- Beginning in Cisco NX-OS Release 9.3(5), a second space is included in the output of the scheduler job configuration CLIs.

  Previously, the output had only one space before the job configuration CLI:

  ```
  scheduler job name show_fds.
   show clock >> bootflash:show_fds
  ^ (single space)
  ```

  Now it has two spaces before the job configuration CLI:

  ```
  scheduler job name show_fds.
    show clock >> bootflash:show_fds
  ^^ (two spaces)
  ```

  There is no impact on the functionality of the scheduler in the NX-OS software for configuration replace, ISSU, reload, and so on. But if you are using a script to read the output of the show run command for reading the scheduler component configuration, then you must update the logic in the script to allow for the extra space.

Enable or Disable the Scheduler

You can enable the scheduler feature so that you can configure and schedule jobs, or you can disable the scheduler feature after it has been enabled.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Enable or disable the scheduler using the command [**no**] **feature scheduler**

Example:

```
switch(config)# feature scheduler
```

Step 3 Display the scheduler configuration using the command **show scheduler config**

Example:

```
switch(config)# show scheduler config
  config terminal
  feature scheduler
```

```
scheduler logfile size 16
end
```

This is an optional step.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Define Scheduler Log File Size

You can configure the log file size for capturing jobs, schedules, and job output.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Define the scheduler log file size in kilobytes using the command **scheduler logfile size** *value*

Example:

```
switch(config)# scheduler logfile size 1024
```

The range is from 16 to 1024. The default is 16.

Note If the size of the job output is greater than the size of the log file, then the output is truncated.

Step 3 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure Remote User Authentication

You can configure the scheduler to use remote authentication for users who want to configure and schedule jobs.

Note Remote users must authenticate with their clear text password before creating and configuring jobs.

Note Remote user passwords are always shown in encrypted form in the output of the **show running-config** command. The encrypted option (**7**) in the command supports the ASCII device configuration.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure a cleartext password for the user who is currently logged in using the command **scheduler aaa-authentication password** [**0** | **7**] *password*

Example:

```
switch(config)# scheduler aaa-authentication password X12y34Z56a
```

Step 3 Configure a cleartext password for a remote user using the command **scheduler aaa-authentication username** *name* **password** [**0** | **7**] *password*

Example:

```
switch(config)# scheduler aaa-authentication username newuser password Z98y76X54b
```

Step 4 Display the scheduler password information using the command **show running-config | include "scheduler aaa-authentication"**

Example:

```
switch(config)# show running-config | include "scheduler aaa-authentication"
```

This is an optional step.

Step 5 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Define Job

You can define a job including the job name and the command sequence.



Caution After you define a job, you cannot modify or remove commands. To change the job, you must delete it and create a new one.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Create a job and enter the job configuration mode using the command **scheduler job name** *string*

Example:

```
switch(config)# scheduler job name backup-cfg
 switch(config-job)
```

This example creates a scheduler job named "backup-cfg".

Step 3 Define the sequence of commands for the specified job using the command *command1* ;[*command2* ; *command3* ;...]

Example:

```
switch(config-job)# copy running-config
 tftp://1.2.3.4/$(SWITCHNAME)-cfg.$(TIMESTAMP) vrf management
 switch(config-job)#
```

Separate commands with spaces and semicolons (for example, " ;").

This example creates a scheduler job that saves the running configuration to a file in the bootflash. The job then copies the file from the bootflash to a TFTP server and creates the filename using the current timestamp and switch name.

Step 4 Display the job information using the command **show scheduler job** [**name** *name*]

Example:

```
switch(config-job)# show scheduler job
```

This is an optional step.

Step 5 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Delete Job

You can delete a job from the scheduler.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Delete the specified job and all commands defined within it using the command **no scheduler job name** *string*

Example:

```
switch(config)# no scheduler job name configsave
 switch(config-job)
```

Step 3 Display the job information using the command **show scheduler job** [**name** *name*]

Example:

```
switch(config-job)# show scheduler job name configsave
```

This is an optional step.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Define Timetable

You can define a timetable in the scheduler to be used with one or more jobs.

If you do not specify the time for the **time** commands, the scheduler assumes the current time. For example, if the current time is March 24, 2013, 22:00 hours, jobs are started as follows:

- For the **time start 23:00 repeat 4:00:00** command, the scheduler assumes a start time of March 24, 2013, 23:00 hours.
- For the **time daily 55** command, the scheduler assumes a start time every day at 22:55 hours.
- For the **time weekly 23:00** command, the scheduler assumes a start time every Friday at 23:00 hours.
- For the **time monthly 23:00** command, the scheduler assumes a start time on the 24th of every month at 23:00 hours.



Note The scheduler will not begin the next occurrence of a job before the last one completes. For example, you have scheduled a job to be completed at one-minute intervals beginning at 22:00; but the job requires two minutes to complete. The scheduler starts the first job at 22:00, completes it at 22:02, and then observes a one-minute interval before starting the next job at 22:03.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Create a new schedule and places you in schedule configuration mode for that schedule using the command **scheduler schedule name** *string*

Example:

```
switch(config)# scheduler schedule name weekendbackupqos
 switch(config-schedule)#
```

Step 3 Associate a job with this schedule using the command **job name** *string*

Example:

```
switch(config-schedule)# job name offpeakZoning
```

You can add multiple jobs to a schedule.

Step 4 Indicate the job starts every day at a designated time specified as HH:MM using the command **time daily** *time*

Example:

```
switch(config-schedule)# time daily 23:00
```

Step 5 Indicate that the job starts on a specified day of the week using the command **time weekly** [[*dow* :] *HH* :] *MM*

Example:

```
switch(config-schedule)# time weekly Sun:23:00
```

Day of the week (dow) specified as one of the following:

- An integer such as 1 = Sunday, 2 = Monday, and so on.
- An abbreviation such as Sun = Sunday.

The maximum length for the entire argument is 10.

Step 6 Indicate the job starts on a specified day each month (dm) using the command **time monthly** [[*dm* :] *HH* :] *MM*

Example:

```
switch(config-schedule)# time monthly 28:23:00
```

If you specify either 29, 30, or 31, the job is started on the last day of each month.

Step 7 Indicate the job starts periodically using the command **time start** { **now repeat** *repeat-interval* | *delta-time* [**repeat** *repeat-interval*]}

Example:

```
switch(config-schedule)# time start now repeat 48:00
```

The start-time format is [[[[yyyy:]mmm:]dd:]HH]:MM.

- *delta-time* —Specifies the amount of time to wait after the schedule is configured before starting a job.
- **now** —Specifies that the job starts now.
- **repeat** *repeat-interval* —Specifies the frequency at which the job is repeated.

In this example, the job starts immediately and repeats every 48 hours.

Step 8 Display the scheduler configuration using the command **show scheduler config**

Example:

```
switch(config)# show scheduler config
```

This is an optional step.

Step 9 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Clear Scheduler Log File

You can clear the scheduler log file.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Clear the scheduler log file using the command **clear scheduler logfile**

Example:

```
switch(config)# clear scheduler
 logfile
```

Verify Scheduler Configuration

To display the scheduler configuration information, perform one of the following tasks:

Command	Purpose
show scheduler config	Displays the scheduler configuration.
show scheduler job [**name** *string*]	Displays the jobs configured.
show scheduler logfile	Displays the contents of the scheduler log file.
show scheduler schedule [**name** *string*]	Displays the schedules configured.

Configuration Examples

Display Job Schedule

This example shows how to display the job schedule:

```
switch# show scheduler schedule
Schedule Name : daily
---------------------------
User Name : admin
Schedule Type : Run every day at 1 Hrs 00 Mins
Last Execution Time : Fri Jan 2 1:00:00 2013
Last Completion Time: Fri Jan 2 1:00:01 2013
Execution count : 2
-----------------------------------------------
Job Name Last Execution Status
-----------------------------------------------
back-cfg Success (0)
switch#
```

Schedule a Scheduler Job

This example shows how to schedule a scheduler job called backup-cfg to run daily at 1 a.m.:

```
switch# configure terminal
switch(config)# scheduler schedule name daily
switch(config-if)# job name backup-cfg
switch(config-if)# time daily 1:00
switch(config-if)# end
switch(config)#
```

Create Scheduler Job

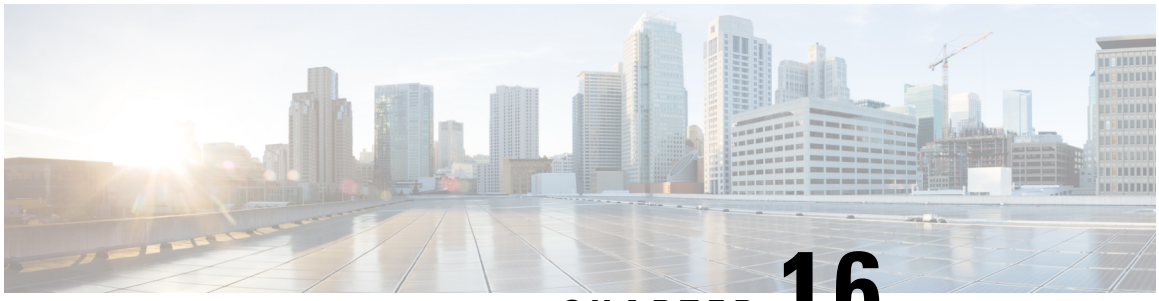
This example shows how to create a scheduler job that saves the running configuration to a file in the bootflash. The job then copies the file from the bootflash to a TFTP server (creates the filename using the current timestamp and switch name):

```
switch# configure terminal
switch(config)# scheduler job name backup-cfg
switch(config-job)# copy running-config
tftp://1.2.3.4/$(SWITCHNAME)-cfg.$(TIMESTAMP) vrf management
switch(config-job)# end
switch(config)#
```

Display Results of Running Scheduler Jobs

This example shows how to display the results of scheduler jobs that have been executed by the scheduler:

```
switch# show scheduler logfile
Job Name : back-cfg Job Status: Failed (1)
Schedule Name : daily User Name : admin
Completion time: Fri Jan 1 1:00:01 2013
-------------------------------- Job Output --------------------------------
`cli var name timestamp 2013-01-01-01.00.00`
`copy running-config bootflash:/$(HOSTNAME)-cfg.$(timestamp)`
`copy bootflash:/switch-cfg.2013-01-01-01.00.00 tftp://1.2.3.4/ vrf management `
copy: cannot access file '/bootflash/switch-cfg.2013-01-01-01.00.00'
==============================================================================
Job Name : back-cfg Job Status: Success (0)
Schedule Name : daily User Name : admin
Completion time: Fri Jan 2 1:00:01 2013
-------------------------------- Job Output --------------------------------
`cli var name timestamp 2013-01-02-01.00.00`
`copy running-config bootflash:/switch-cfg.2013-01-02-01.00.00`
`copy bootflash:/switch-cfg.2013--01-02-01.00.00 tftp://1.2.3.4/ vrf management `
Connection to Server Established.
[ ] 0.50KBTrying to connect to tftp server......
[###### ] 24.50KB
TFTP put operation was successful
==============================================================================
switch#
```



CHAPTER 16

Simple Network Management Protocol

The Simple Network Management Protocol (SNMP) is an application-layer protocol that provides a message format for communication between SNMP managers and agents. SNMP provides a standardized framework and a common language used for the monitoring and management of devices in a network.

Note The license notifications are enabled by default.

- SNMP Fundamentals, on page 243
- Configure SNMP, on page 253
- Verify SNMP Configuration, on page 279
- Configuration Examples, on page 280
- SNMP Sample Outputs, on page 282
- Related Documents, on page 283

SNMP Fundamentals

Framework

The SNMP framework consists of three parts:

- An SNMP manager—The system used to control and monitor the activities of network devices using SNMP.
- An SNMP agent—The software component within the managed device that maintains the data for the device and reports these data, as needed, to managing systems. The Cisco Nexus device supports the agent and MIB. To enable the SNMP agent, you must define the relationship between the manager and the agent.
- A managed information base (MIB)—The collection of managed objects on the SNMP agent

Note SNMP is defined in RFCs 3411 to 3418. The device supports SNMPv1, SNMPv2c, and SNMPv3. Both SNMPv1 and SNMPv2c use a community-based form of security.

Cisco NX-OS supports SNMP over IPv6.

Notifications

A key feature of SNMP is the ability to generate notifications from an SNMP agent. These notifications do not require that requests be sent from the SNMP manager. Notifications can indicate improper user authentication, restarts, the closing of a connection, loss of connection to a neighbor router, or other significant events.

NX-OS generates SNMP notifications as either traps or informs. A trap is an asynchronous, unacknowledged message sent from the agent to the SNMP managers listed in the host receiver table. Informs are asynchronous messages sent from the SNMP agent to the SNMP manager which the manager must acknowledge receipt of.

Traps are less reliable than informs because the SNMP manager does not send any acknowledgment when it receives a trap. The device cannot determine if the trap was received. An SNMP manager that receives an inform request acknowledges the message with an SNMP response protocol data unit (PDU). If the device never receives a response, it can send the inform request again.

You can configure NX-OS to send notifications to multiple host receivers.

The following table lists the SNMP traps that are enabled by default.

Trap Type	Description
generic	: coldStart
entity	: entity_fan_status_change
entity	: entity_mib_change
entity	: entity_module_status_change
entity	: entity_module_inserted
entity	: entity_module_removed
entity	: entity_power_out_change
entity	: entity_power_status_change
entity	: entity_unrecognised_module
link	: cErrDisableInterfaceEventRev1
link	: cieLinkDown
link	: cieLinkUp
link	: cmn-mac-move-notification
link	: delayed-link-state-change
link	: extended-linkDown
link	: extended-linkUp

Trap Type	Description
link	: linkDown
link	: linkUp
rf	: redundancy_framework
upgrade	: UpgradeOpNotifyOnCompletion
upgrade	: UpgradeJobStatusNotify
entity	: entity_sensor
rmon	: fallingAlarm
rmon	: hcRisingAlarm
rmon	: hcFallingAlarm
rmon	: risingAlarm

Security Levels and Models

A security model is an authentication strategy that is set up for a user and the role in which the user resides. A security level is the permitted level of security within a security model. A combination of a security model and a security level determines which security mechanism is employed when handling an SNMP packet. SNMPv3 provides for both security models and security levels.

The security level determines if an SNMP message needs to be protected from disclosure and if the message needs to be authenticated. The various security levels that exist within a security model are as follows:

- noAuthNoPriv—Security level that does not provide authentication or encryption. This level is not supported for SNMPv3.
- authNoPriv—Security level that provides authentication but does not provide encryption.
- authPriv—Security level that provides both authentication and encryption.

The security models available are:

- SNMPv1
- SNMPv2c
- SNMPv3

The following table identifies what the combinations of security models and levels mean.

Table 20: SNMP Security Models and Levels

Model	Level	Authentication	Encryption	What Happens
v1	noAuthNoPriv	Community string	No	Uses a community string match for authentication.

Model	Level	Authentication	Encryption	What Happens
v2c	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
v3	authNoPriv	HMAC-MD5, HMAC-SHA, or SHA-256	No	Provides authentication based on the Hash-Based Message Authentication Code (HMAC) Message Digest 5 (MD5) algorithm or the HMAC Secure Hash Algorithm (SHA).
v3	authPriv	HMAC-MD5, HMAC-SHA, or SHA-256	DES or AES-128	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides Data Encryption Standard (DES) 56-bit encryption in addition to authentication based on the Cipher Block Chaining (CBC) DES (DES-56) standard.

SNMP Logical Entities

A device can support multiple instances of a logical network entity, such as protocol instances or virtual routing and forwarding (VRF) instances. Most existing MIBs cannot distinguish between the multiple logical network entities. For example, the original OSPF-MIB assumes a single protocol instance on a device, but you can now configure multiple OSPF instances on a device.

Context

An SNMP context is a collection of management information that you can access through the SNMP agent.

Groups

SNMP access rights are organized by groups. Each group in SNMP is similar to a role through the CLI. Each group is defined with read access or read-write access.

You can begin communicating with the agent once your username is created, your roles are set up by your administrator, and you are added to the roles.

Note Because *group* is a standard SNMP term used industry-wide, we refer to roles as groups in this SNMP section.

Embedded Event Manager (EEM)

The Embedded Event Manager (EEM) feature monitors events, including SNMP MIB objects, and triggers an action based on these events. One of the actions could be to send an SNMP notification. EEM sends the cEventMgrPolicyEvent of CISCO-EMBEDDED-EVENT-MGR-MIB as the SNMP notification.

VRF Support

SNMP is also VRF aware. You can configure SNMP to use a particular VRF to reach the SNMP notification host receiver. You can also configure SNMP to filter notifications to an SNMP host receiver based on the VRF where the notification occurred.

Cisco NX-OS Support

Cisco NX-OS supports one instance of the SNMP. Cisco NX-OS supports stateless restarts for SNMP. After a reboot or supervisor switchover, Cisco NX-OS applies the running configuration. Cisco NX-OS supports SNMP over IPv6.

Multiple Instance Support

SNMP supports multiple MIB module instances and maps them to logical network entities. Cisco NX-OS supports the CISCO-CONTEXT-MAPPING-MIB to map between SNMP contexts and logical network entities. You can associate an SNMP context to a VRF, protocol instance, or topology.

SNMPv3 - uses contexts to distinguish between these multiple instances. A device can support multiple contexts for different logical network entities. An SNMP context allows the SNMP manager to access one of the multiple instances of a MIB module supported on the device for the different logical network entities. SNMPv3 supports contexts with the contextName field of the SNMPv3 PDU. You can map this contextName field to a particular protocol instance or VRF.

SNMPv2c - you can map the SNMP community to a context using the snmpCommunityContextName MIB object in the SNMP-COMMUNITY-MIB (RFC 3584). You can then map this snmpCommunityContextName to a particular protocol instance or VRF using the CISCO-CONTEXT-MAPPING-MIB or the CLI.

Note SNMP is defined in RFCs 3411 to 3418. The device supports SNMPv1, SNMPv2c, and SNMPv3. Both SNMPv1 and SNMPv2c use a community-based form of security.

SNMPv3

SNMPv3 provides secure access to devices by a combination of authenticating and encrypting frames over the network. The SNMPv3 User-Based Security Model (USM) refers to SNMP message-level security and offers the following services:

- Message integrity— Ensures that messages have not been altered or destroyed in an unauthorized manner and that data sequences have not been altered to an extent greater than can occur nonmaliciously.
- Authentication—Ensures that the claimed identity of the user on whose behalf received data was originated is confirmed.

- Encryption—Ensures that information is not made available or disclosed to unauthorized individuals, entities, or processes.

SNMPv3 authorizes management operations only by configured users and encrypts SNMP messages.

Authentication Protocols

NX-OS uses three authentication protocols for SNMPv3:

- HMAC-MD5-96 authentication protocol
- HMAC-SHA-96 authentication protocol
- SHA-256 authentication protocol



Note Beginning with Cisco NX-OS release 9.3(7), HMAC-SHA-256 authentication protocol is used for SNMPv3. When SHA-256 SNMP users are configured on the switch, ISSD is recommended by **install all** cmd else there will be config loss.

NX-OS uses Advanced Encryption Standard (AES) as one of the privacy protocols for SNMPv3 message encryption and conforms with RFC 3826.

The **priv** option offers a choice of DES or 128-bit AES encryption for SNMP security encryption. The **priv** option and the **aes-128** token indicate that this privacy password is for generating a 128-bit AES key. The AES priv password can have a minimum of eight characters. If the passphrases are specified in clear text, you can specify a maximum of 64 case-sensitive, alphanumeric characters. If you use the localized key, you can specify a maximum of 130 characters.



Note SNMPWALK with remote user will only work if user has been created as SNMP user. DCNM/ND is the only NMS which creates SNMP remote user. None of the remote or external AAA server directly creates SNMP remote user.

User Configuration Synchronization

SNMPv3 user management can be centralized at the Access Authentication and Accounting (AAA) server level. This centralized user management allows the SNMP agent in NX-OS to leverage the user authentication service of the AAA server. Once user authentication is verified, the SNMP PDUs are processed further. Additionally, the AAA server is also used to store user group names. SNMP uses the group names to apply the access/role policy that is locally available in the switch.

Any configuration changes made to the user group, role, or password results in database synchronization for both SNMP and AAA.

NX-OS synchronizes the user configuration in the following ways:

- The authentication passphrase specified in the **snmp-server user** command becomes the password for the CLI user.
- The password specified in the **username** command becomes the authentication and privacy passphrases for the SNMP user.

- If you create or delete a user using either SNMP or the CLI, the user is created or deleted for both SNMP and the CLI.
- User-role mapping changes are synchronized in SNMP and the CLI.
- Role changes (deletions or modifications) from the CLI are synchronized to SNMP.

Note When you configure a passphrase/password in localized key/encrypted format, NX-OS does not synchronize the user information (passwords, roles, and so on). Cisco NX-OS holds the synchronized user configuration for 60 minutes by default.

Disable Security and SNMP User Synchronization

Beginning with Cisco NX-OS Release 10.2(2)F, the following desynchronization command is introduced to provide you an option to disable the user synchronization between the SNMP and the security (AAA or CLI) components:

snmp-server disable snmp-aaa sync

You can execute this command from the configure terminal on the Nexus switches. By default, the **no** form of the desynchronization command is available on the switch.

When the no-form of the desynchronization command is enabled on the device, for example, `switch (config)#` **no snmp-server disable snmp-aaa sync**, a user created through **snmp-server user** CLI results in the creation of a **username** CLI for that user in the running configuration and conversely. So, the user can log in to the switch, using the authentication credentials mentioned in the **snmp-server user** CLI or the **username** CLI, at the time of creation/updation, and will also be able to perform SNMP operations from a network manager on the switch. Thus, the **no** form of the desynchronization command ensures that the user synchronization between the SNMP and the AAA functions the way it did in the releases prior to 10.2(2)F.

When the desynchronization command is enabled on the device, for example, `switch (config)#` **snmp-server disable snmp-aaa sync**, a user created through the **snmp-server user** command does not create a username configuration for that user. So, the user cannot log in to the switch and is only allowed to do SNMP operations through a network manager on the switch. Similarly, creation of a security user through the **username** CLI does not create a corresponding **snmp-server user** CLI for the user. This user will be able to log in to the switch but will not be able to perform any SNMP operation on the switch. This is a new feature that the desynchronization command has introduced from Release 10.2(2)F.

You can view the status of the desynchronization command in one of the following ways:

- The value of the field `SNMP-AAA sync disable` in the output of the CLI **show snmp internal globals**
- The value of the field `disableSnmpAaaSync` in the `sys/snmp/inst/globals` MO
- The CLI print in the **show-running-config** output and **show-running-config-snmp** output or **show-running-all** output, based on whether the command is enabled or disabled, respectively

Remote Users

With regard to remote users, who are authenticated for login through external servers using protocols such as RADIUS and TACACS+, when the desynchronization command is enabled on the switch, the remote users cannot be created in SNMP. For more information, refer to the *Configuring AAA* chapter in the *Cisco Nexus 9000 NX-OS Security Configuration Guide*.

However, when the **no** form of the desynchronization command is enabled on the switch, if a remote user is created in AAA, the corresponding user is created in SNMP as well. Furthermore, the user will not be available in the running-config output of SNMP, but will be able to perform SNMP operations on the managed device, which is an existing feature prior to Release 10.2(2)F. However, external NMS client's originated snmpwalk and so on does not go through unless the **snmp-sync-password** command is executed in the device for the remote user created in AAA.

DCNM Security Users

The security users created using DCNM (also called as Nexus Dashboard Fabric Controller from Release 12.0.1a onwards) will not have a corresponding SNMPv3 profile when the desynchronization command is enabled. When the synchronization is disabled, the users created on the security component can log in to the switch, but the switches will not be discovered by the controller, as the controller uses the SNMP configuration created for the security user to discover the switch. Furthermore, the SNMP does not recognize the security users created due to the desynchronized state of the userDB, resulting in failure to discover the switch. Therefore, to have the switches discovered by the controller, the SNMP user must be explicitly created. It is not recommended to use the desynchronization command along with DCNM functionality. For more information, refer to the *Cisco Nexus 9000 NX-OS Security Configuration Guide*.

ISSD and ISSU

In general, if SNMP user synchronization has been disabled, do not enable SNMP user synchronization unless all the desynchronized users are removed. A running configuration with such a combination will result in a configuration replace failure.

The only way to achieve the desynchronized state in older releases without the desynchronization command is as follows:

- If the Disruptive/ND-ISSD is performed from a desynchronized state to a release without the desynchronization command, the desynchronized databases will be ported as-is through ISSD to the previous release.



Note Any modifications done to the user database after such ISSD will be synchronized between SNMP and security components.

After such ISSD, ISSU to a release with desynchronization command brings in the desynchronized user database as-is, but the desynchronization command comes up in its default **no** form. If required, enable the desynchronization command.

Guidelines and Limitations

SNMP has the following configuration guidelines and limitations:

- Commands configured using SNMP SET should be deleted using SNMP SET only. Commands configured using Command Line Interface(CLI) or NX-API should be deleted using CLI or NX-API only.
- When you create or edit a user in AAA using clear text password, SNMP creates or edits the user to have default auth (md5) and priv types.

  When you create or edit a user in SNMP using clear text password, AAA creates or edits the user to have default password type (type 5).

- Access control list (ACLs) can be applied only to local SNMPv3 users configured on the switch. ACLs cannot be applied to remote SNMPv3 users stored on Authentication, Authorization, and Accounting (AAA) servers.
- Do not enable SNMP user synchronisation after it has been disabled unless all desynchronised users are removed. A running configuration with such a combination will result in a configuration replace failure.
- Cisco NX-OS supports read-only access to some SNMP MIBs. See the Cisco NX-OS MIB support list at the following URL for more information: https://cisco.github.io/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html
- When VLAN 1 is not allowed on a trunk, the global SNMP context might not return `BRIDGE-MIB::dot1dBasePort` and `BRIDGE-MIB::dot1dBasePortIfIndex` mappings for that trunk. To retrieve the mappings, poll `BRIDGE-MIB` in the SNMPv3 context for the relevant VLAN, or use `CISCO-IF-EXTENSION-MIB::cieIfDot1dBaseMappingPort` and reverse the mapping in your network-management application.
- Cisco NX-OS does not support the SNMPv3 noAuthNoPriv security level.
- Cisco Nexus 9000 Series switches and the Cisco Nexus 3164Q, 31128PQ, 3232C, and 3264Q switches support the configuration of the SNMP local engine ID.
- For a nondisruptive downgrade path to an earlier release, if a local engine ID has been configured, then you must unconfigure the local engine ID, and then reconfigure the SNMP users and the community strings.
- Special characters @ and % are not allowed in the SNMP community string.
- The default SNMP PDU value is 1500 bytes. The SNMP agent drops any response PDU that is greater than 1500 bytes, causing the SNMP request to fail. To receive MIB data values larger than 1500 bytes, use the **snmp-server packetsize** <*byte-count*> command to reconfigure the packet size. The valid byte-count range is from 484 to 17382. When a GETBULK response exceeds the packet size, the data can get truncated.
- You must use either the CLI or SNMP to configure a feature on your switch. Do not configure a feature using both interfaces to the switch.
- Using cefcFanTrayOperStatus snmpwalk on an individual fan OID tree where the fan is not populated in chassis, can return a response for next OID entry in the tree. To prevent this behavior, use the -CI option in *snmpwalk*.

  The behavior is not seen when polling parent OID, or when using getmany.
- Cisco Nexus 9000 series switches support upto 20000 flash files for *snmpwalk* request.
- There must be at least one running BGP instance to have full, proper functional behavior of SNMP traps. Configure a BGP routing instance before configuring any snmp-server traps related commands.
- Beginning with Release 10.1(1), AES-128 is the recommended encryption algorithm, as it is a strong encryption algorithm. However, DES encryption is also supported.

  Downgrade: In-Service System Downgrade (ISSD) with **install all** command is aborted if users with DES privacy protocol are present in the SNMP database. Users need to be reconfigured (using the default AES-128) or deleted. In case of a cold reboot, the SNMP users with DES are deleted.
- Beginning with Cisco NX-OS Release 10.5(2), users can configure AES-256 as the privacy protocol for SNMPv3.

- Before downgrading to the earlier releases, reconfigure the existing user with encryption AES-256 to AES-128 or remove the user with encryption AES-256.
- This feature is supported on all N9K platforms.

- When engine ID is configured after configuring the SNMP user, ensure that you perform the following action:
  - After changing the engine ID, reconfigure the SNMP user and the related configuration including group, ACL, along with the password. This avoids authentication failure and impact on the ACL and group attached to the user.
- Beginning with Cisco NX-OS Release 10.3(1)F, SNMP (MIBs – 400G Optic MIB, Switch MIB, Datapath MIBs, Interface MIB) is supported on the Cisco Nexus 9808 platform switches.
- The SVI stats are polled only at an interval of every 120 seconds for SNMP cache.
- Beginning with Cisco NX-OS Release 10.3(3)F, Type-6 encryption for SNMPv3 user password is supported with following limitations:
  - Type-6 encryption is successful only if the following is taken care:
    - **feature password encryption aes {tam}** is enabled.
    - Primary key is configured.
    - The **pwd_type 6** option is specified during SNMPv3 user configuration.
  - Changing the primary key configuration results in SNMP re-encrypting all Type-6 users stored in its database. However, the SNMP functionalities continue to work the same way as before.
  - Primary key configuration is local to the switch. If the user takes the Type-6 configured running data from one switch and applies it on other switch where a different primary key is configured, SNMP features for the same user might not work on the other switch.
  - If Type-6 is configured, ensure to remove the configuration, or reconfigure the Type-6 option before downgrading to the release where Type-6 is not supported.
  - In case of ISSU, if you migrate from an earlier image (where localizedkey, localizedV2key config is present) to a new image where Type-6 encryption is supported, SNMP won't convert the existing keys to Type-6 encryption.
  - Conversion between existing SALT encryption to Type-6 encryption is supported using the **encryption re-encrypt obfuscated** command.
  - ASCII-based reloads through disruptive upgrades and **reload-ascii** commands leads to loss of primary key which would impact the SNMP functionality for the Type-6 users.
  - If a user enforces re-encryption using the **encryption re-encrypt obfuscated** command, then SNMP encrypts all passwords from non-Type-6 SNMP users to Type-6 mode.



Note The SNMP does not support the **encryption delete type6** command.

- From Cisco NX-OS Release 10.4(1)F, you can view Electronic Programmable Logic Device (EPLD) firmware version using SNMP. As part of Entity MIB structure, you can view the firmware version, type of EPLD device (IO or MI FPGA), and the parent entity of the EPLD devices, such as supervisor, line card, and Line card expansion module (LEM). This feature is supported for Cisco Nexus 9300-FX/FX2/FX3/GX platform switches, N9K-C9332D-H2R switch, and Nexus 9508 switch. Beginning with Cisco NX-OS Release 10.4(3)F, this feature is also supported on N9K-C9364C-H1 switch.Beginning with Cisco NX-OS Release 10.5(2)F, this feature is also supported on N9K-X9736C-FX3 line card.
- Beginning with Cisco NX-OS Release 10.4(1)F, SNMP (MIBs – 400G Optic MIB, Switch MIB, Datapath MIBs, Interface MIB) is supported on the following line cards and switches:
  - Cisco Nexus 9804 switch
  - Cisco Nexus C9332D-H2R switch
  - Cisco Nexus X98900CD-A and X9836DM-A line cards with Cisco Nexus 9808 and 9804 switches
- Beginning with Cisco NX-OS Release 10.4(2)F, SNMP (MIBs – 400G Optic MIB, Switch MIB, Datapath MIBs, Interface MIB) is supported on Cisco Nexus 93400LD-H1 platform switches.
- Beginning with Cisco NX-OS Release 10.4(2)F, SNMP (MIBs – 400G Optic MIB, Switch MIB, Datapath MIBs, Interface MIB) is supported on Cisco Nexus N9K-C9364C-H1 platform switches.
- Beginning with NX-OS Release 10.6(2n), SNMP is supported on N9164E-NS4-O switch.

Cisco N9300 Smart Switches

- Beginning with Cisco NX-OS Release 10.6(2F), SNMPv1, SNMPv2c, and SNMPv3 are supported on the Cisco N9324C-SE1U and N9348Y2C6D-SE1U switches.

Configure SNMP

Note Be aware that the Cisco NX-OS commands for this feature may differ from those commands used in Cisco IOS.

Note From Cisco NX-OS release 9.3(7), HMAC-SHA-256 authentication protocol is used for SNMPv3.

The following section includes the various processes involved to configure SNMP.

Configure SNMP Users

You can configure a user for SNMP.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure an SNMP user with authentication and privacy parameters using the command **snmp-server user***name* [**pwd_type 6**] [**auth** { **md5** | **sha** | **sha-224** | **sha-256** | **sha-384** | **sha-512** } *passphrase* [**auto**] [**priv[aes-128][aes-256]***passphrase*] [**engineID***id*] [**localizedkey**] | [**localizedV2key**]]

Example:

```
switch(config)# snmp-server user Admin pwd_type 6 auth sha abcd1234 priv abcdefgh
```

The passphrase can be any case-sensitive, alphanumeric string up to 64 characters. If you use the **localizedkey** keyword, the passphrase can be any case-sensitive, alphanumeric string up to 130 characters.

sha - Use the HMAC SHA-1 algorithm for authentication.

sha-224 - Use the HMAC SHA-224 algorithm for authentication.

sha-256 - Use the HMAC SHA-256 algorithm for authentication.

sha-384 - Use the HMAC SHA-384 algorithm for authentication.

sha-512 - Use the HMAC SHA-512 algorithm for authentication.

localizedkey - If you use the **localizedkey** keyword, the passphrase can be any case-sensitive, alphanumeric string up to 130 characters. Instead of plain-text password, **hashed** password (copied either from the **show running config** command or generated offline using snmpv3 based open source hash generator tool, see Generate Hashed Password Offline, on page 255) can be configured using the localizedkey keyword.

Note When using a localized key, add `0x` before the hash value, for example, `0x84a716329158a97ac9f22780629bc26c` .

localizedV2key - If the localizedV2key is used, the passphrase can be any case-sensitive, alphanumeric string up to 130 characters, without 0x at the beginning. Collect the localizedv2key using **show run** command, as this is an encrypted data and cannot be generated offline.

The engineID format is a 12-digit, colon-separated decimal number.

Note

- Beginning with Cisco NX-OS Release 10.1(1), AES-128 is the default privacy protocol for SNMPv3.
- Beginning with Cisco NX-OS Release 10.5(2), users can configure AES-256 as the privacy protocol for SNMPv3.
- Beginning with Cisco NX-OS Release 10.3(3)F, the **pwd_type 6** keyword is supported to provide Type-6 encryption for SNMP users password.

Step 3 Display information about one or more SNMP users using the command **show snmp user**

Example:

```
switch(config)# show snmp user
```

This is an optional step.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Generate Hashed Password Offline

Perform the following steps to generate hashed password offline, using snmpv3-based open source hash generator tool:

Note The IDs mentioned in this procedure are only sample IDs, the purpose of which is only to explain the procedure better.

1. Get the SNMP engineID from the switch.

   switch# **show snmp engineID**

   Sample output:

   ```
   Local SNMP engineID: [Hex] 8000000903D4C93CEA31CC
   [Dec] 128:000:000:009:003:212:201:060:234:049:204
   ```

2. Use an SNMPv3 based open source hash generator to generate offline hashed password.

   Linux$ **snmpv3-hashgen --auth** Hello123 --engine 8000000903D4C93CEA31CC --user1 **--mode priv --hash md5**

   Sample output:

   ```
   User: user1
   Auth: Hello123 / 84a716329158a97ac9f22780629bc26c
   Priv: Hello123 / 84a716329158a97ac9f22780629bc26c
   Engine: 8000000903D4C93CEA31CC
   ESXi USM String: u1/84a716329158a97ac9f22780629bc26c/84a716329158a97ac9f22780629bc26c/priv
   ```

3. Use the auth and priv values to configure the password on the switch.

   snmp-server user user1 **auth md5** 0x84a716329158a97ac9f22780629bc26c **priv des 0x**84a716329158a97ac9f22780629bc26c **localizedkey**

Enforce SNMP Message Encryption

You can configure SNMP to require authentication or encryption for incoming requests. By default, the SNMP agent accepts SNMPv3 messages without authentication and encryption. When you enforce privacy, NX-OS

responds with an authorization error for any SNMPv3 PDU request using a security level parameter of either **noAuthNoPriv** or **authNoPriv** .

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Enforce SNMP message encryption for this user using the command **snmp-server user***name***enforcePriv**

Example:

```
switch(config)# snmp-server user Admin enforcePriv
```

Step 3 Enforce SNMP message encryption for all users using the command **snmp-server globalEnforcePriv**

Example:

```
switch(config)# snmp-server globalEnforcePriv
```

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Assign SNMPv3 Users to Multiple Roles

After you configure an SNMP user, you can assign multiple roles for the user.



Note Only users belonging to a network-admin role can assign roles to other users.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Associate this SNMP user with the configured user role using the command **snmp-server user***namegroup*

Example:

```
switch(config)# snmp-server user Admin superuser
```

Step 3 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Create SNMP Communities

You can create SNMP communities for SNMPv1 or SNMPv2c.

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Create an SNMP community string using the command **snmp-server community***name* { **group***group* | **ro** | **rw** }

Example:

```
switch(config)# snmp-server community public ro
```

Step 3 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Filter SNMP Requests

You can assign an access control list (ACL) to an SNMPv2 community to filter SNMP requests. If the assigned ACL allows the incoming request packet, SNMP processes the request. If the ACL denies the request, SNMP drops the request and sends a system message.

Create the ACL with the following parameters:

- Source IP address

- Destination IP address
- Source port
- Destination port
- Protocol (UDP or TCP)

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Assign an IPv4 ACL to an SNMPv2 community to filter SNMP requests using the command **snmp-server community***name* [**use-ipv4acl***acl-name*]

Example:

```
switch(config)# snmp-server community public use-ipv4acl myacl
```

Step 3 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure SNMP Notification Receivers

You can configure NX-OS to generate SNMP notifications to multiple host receivers.

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure a host receiver for SNMPv1 traps using the command **snmp-server host***ip-address***traps version 1***community* [**udp_port***number*]

Example:

```
switch(config)# snmp-server host 192.0.2.1 traps version 1 public
```

The *ip-address* can be an IPv4 or IPv6 address. The *community* can be any alphanumeric string up to 255 characters. The UDP port number range is from 0 to 65535.

Step 3 Configure a host receiver for SNMPv2c traps or informs using the command **snmp-server host***ip-address* { **traps** | **informs** } **version 2c***community* [**udp_port***number*]

Example:

```
switch(config)# snmp-server host 192.0.2.1 informs version 2c public
```

The *ip-address* can be an IPv4 or IPv6 address. The *community* can be any alphanumeric string up to 255 characters. The UDP port number range is from 0 to 65535.

Step 4 Configure a host receiver for SNMPv3 traps or informs using the command **snmp-server host***ip-address* { **traps** | **informs** } **version 3** { **auth** | **noauth** | **priv** } *username* [**udp_port***number*]

Example:

```
switch(config)# snmp-server host 192.0.2.1 informs version 3 auth NMS
```

The *ip-address* can be an IPv4 or IPv6 address. The *username* can be any alphanumeric string up to 255 characters. The UDP port number range is from 0 to 65535.

Note The SNMP manager must know the user credentials (authKey/PrivKey) based on the SNMP engine ID of the Cisco NX-OS device to authenticate and decrypt the SNMPv3 messages.

Step 5 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure Source Interface for SNMP Notifications

You can configure SNMP to use the IP address of an interface as the source IP address for notifications. When a notification is generated, its source IP address is based on the IP address of this configured interface.

You can configure a source interface as follows:

- All notifications sent to all SNMP notification receivers.
- All notifications sent to a specific SNMP notification receiver. This configuration overrides the global source interface configuration.

Note Configuring the source interface IP address for outgoing trap packets does not guarantee that the device will use the same interface to send the trap. The source interface IP address defines the source address inside of the SNMP trap, and the connection is opened with the address of the egress interface as source.

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Send traps messages to this host using the command **snmp-server host***ip-address***source-interface***if-type if-number***traps version 2c***name*

Example:

```
snmp-server host 192.0.2.1 source-interface ethernet 2/1 traps version 2c public
```

This is an optional step. The traps version is the SNMP version to use for notification messages. 2c indicates that SNMPv2c is to be used.

Step 3 Configure SNMP to use the selected VRF to communicate with the host receiver using the command **snmp-server host***ip-address***source-interface***if-type if-number***use-vrf***vrf-name*

Example:

```
snmp-server host 192.0.2.1 source-interface ethernet 2/1 use-vrf default
```

The ip-address can be an IPv4 or IPv6 address. The VRF name can be any alphanumeric string up to 32 characters.

Note This command does not remove the host configuration.

Step 4 Configure a host receiver for SNMPv2c traps or informs using the command **snmp-server host***ip-address***source-interface***if-type if-number* [**udp_port***number*]

Example:

```
switch(config)# snmp-server host 192.0.2.1 source-interface ethernet 2/1
```

The *ip-address* can be an IPv4 or IPv6 address. Use ? to determine the supported interface types. The UDP port number range is from 0 to 65535. This configuration overrides the global source interface configuration.

Step 5 Configure a source interface for sending out SNMPv2c traps or informs using the command **snmp-server source-interface** { **traps** | **informs** } *if-type if-number*

Example:

```
switch(config)# snmp-server source-interface traps ethernet 2/1
```

Use ? to determine the supported interface types.

Step 6 Display information about configured source interfaces using the command **show snmp source-interface**

Example:

```
switch(config)# show snmp source-interface
```

Configure Notification Target User

You must configure a notification target user on the device to send SNMPv3 inform notifications to a notification host receiver.

Cisco NX-OS uses the credentials of the notification target user to encrypt the SNMPv3 inform notification messages to the configured notification host receiver.

Note For authenticating and decrypting the received inform PDU, the notification host receiver should have the same user credentials as configured in Cisco NX-OS to authenticate and decrypt the informs.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure the notification target user with the specified engine ID for the notification host receiver using the command **snmp-server user***name* [**auth** { **md5** | **sha** | **sha-256** } *passphrase* [**auto**] [**priv***passphrase*] [**engineID***id*]

Example:

```
switch(config)# snmp-server user NMS auth sha abcd1234 priv abcdefgh engineID
00:00:00:63:00:01:00:10:20:15:10:03
```

The engine ID format is a 12-digit colon-separated decimal number.

Note Beginning with Release 10.1(1), AES-128 is the default privacy protocol for SNMPv3.

Step 3 Copy the running configuration to the startup configuration using the command**copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure SNMP Notification Receivers with VRFs

SNMP adds entries into the cExtSnmpTargetVrfTable of the CISCO-SNMP-TARGET-EXT-MIB when you configure the VRF reachability and filtering options for an SNMP notification receiver.

Note You must configure the host before configuring the VRF reachability or filtering options.

You can configure Cisco NX-OS to use a configured VRF to reach the host receiver or to filter notifications based on the VRF in which the notification occurred.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure SNMP to use the selected VRF to communicate with the host receiver using the command [**no**] **snmp-server host***ip-address***use-vrf***vrf-name* [**udp_port***number*]

Example:

```
switch(config)# snmp-server host 192.0.2.1 use-vrf Blue
```

The *ip-address* can be an IPv4 or IPv6 address. The VRF name can be any alphanumeric string up to 255 characters. The UDP port number range is from 0 to 65535. This command adds an entry into the ExtSnmpTargetVrfTable of the CISCO-SNMP-TARGET-EXT-MB.

The **no** form of this command removes the VRF reachability information for the configured host and removes the entry from the ExtSnmpTargetVrfTable of the CISCO-SNMP-TARGET-EXT-MB.

Note This command does not remove the host configuration.

Step 3 Filter notifications to the notification host receiver based on the configured VRF using the command [**no**] **snmp-server host***ip-address***filter-vrf***vrf-name* [**udp_port***number*]

Example:

```
switch(config)# snmp-server host 192.0.2.1 filter-vrf Red
```

The *ip-address* can be an IPv4 or IPv6 address. The VRF name can be any alphanumeric string up to 255 characters. The UDP port number range is from 0 to 65535.

This command adds an entry into the ExtSnmpTargetVrfTable of the CISCO-SNMP-TARGET-EXT-MB.

The **no** form of this command removes the VRF filter information for the configured host and removes the entry from the ExtSnmpTargetVrfTable of the CISCO-SNMP-TARGET-EXT-MB.



Note This command does not remove the host configuration.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure SNMP to Send Traps Using an Inband Port

You can configure SNMP to send traps using an inband port. To do so, you must configure the source interface (at the global or host level) and the VRF used to send the traps.

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Globally configure a source interface for sending out SNMP traps using the command **snmp-server source-interface traps***if-type if-number*

Example:

```
switch(config)# snmp-server source-interface traps ethernet 1/2
```

Use ? to determine the supported interface types. You can configure the source interface at the global level or at a host level. When the source interface is configured globally, any new host configuration uses the global configuration to send the traps.



Note To configure a source interface at the host level, use the **snmp-server host***ip-address***source-interface***if-type if-number* command.

Step 3 (Optional) Display information about configured source interfaces using the command **show snmp source-interface**

Example:

```
switch(config)# show snmp source-interface
```

Step 4 Configure SNMP to use the selected VRF to communicate with the host receiver using the command **snmp-server host***ip-address***use-vrf***vrf-name* [**udp_port***number*]

Example:

```
switch(config)# snmp-server host 171.71.48.164 use-vrf default
```

The *ip-address* can be an IPv4 or IPv6 address. The VRF name can be any alphanumeric string up to 255 characters. The UDP port number range is from 0 to 65535. This command adds an entry into the ExtSnmpTargetVrfTable of the CISCO-SNMP-TARGET-EXT-MB.

Note By default, SNMP sends the traps using the management VRF. If you do not want to use the management VRF, you must use this command to specify the desired VRF.

Step 5 Display information about configured SNMP hosts using the command **show snmp host**

Example:

```
switch(config)# show snmp host
```

This is an optional step.

Step 6 Copy the running configuration to the startup configuration using the command**copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Enable SNMP Notifications

You can enable or disable notifications. If you do not specify a notification name, Cisco NX-OS enables all notifications except BGP, EIGRP, and OSPF notifications.

Note The **snmp-server enable traps** command enables both traps and informs, depending on the configured notification host receivers.

The following table lists the commands that enable the notifications for NX-OS MIBs.

Table 21: Enabling SNMP Notifications

MIB	Related Commands
All notifications (except BGP, EIGRP, and OSPF)	**snmp-server enable traps**
CISCO-AAA-SERVER-MIB	**snmp-server enable traps aaa** **snmp-server enable traps aaa server-state-change**
CISCO-BGP4-MIB	**snmp-server enable traps bgp**
CISCO-CALLHOME-MIB	**snmp-server enable traps callhome** **snmp-server enable traps callhome event-notify** **snmp-server enable traps callhome smtp-send-fail**
CISCO-CONFIG-MAN-MIB	**snmp-server enable traps config** **snmp-server enable traps config ccmCLIRunningConfigChanged**
CISCO-EIGRP-MIB	**snmp-server enable traps eigrp** [*tag*]
CISCO-ERR-DISABLE-MIB	**snmp-server enable traps link cerrDisableInterfaceEventRev1**
ENTITY-MIB, CISCO-ENTITY-SENSOR-MIB	**snmp-server enable traps entity** **snmp-server enable traps entity entity_fan_status_change** **snmp-server enable traps entity entity_mib_change** **snmp-server enable traps entity entity_module_inserted** **snmp-server enable traps entity entity_module_removed** **snmp-server enable traps entity entity_module_status_change** **snmp-server enable traps entity entity_power_out_change** **snmp-server enable traps entity entity_power_status_change** **snmp-server enable traps entity entity_unrecognised_module**
CISCO-FEATURE-CONTROL-MIB	**snmp-server enable traps feature-control** **snmp-server enable traps feature-control FeatureOpStatusChange**

MIB	Related Commands
CISCO-HSRP-MIB	**snmp-server enable traps hsrp** **snmp-server enable traps hsrp state-change**
IF-MIB	**snmp-server enable traps link** **snmp-server enable traps link extended-linkDown** **snmp-server enable traps link extended-linkUp** **snmp-server enable traps link cieLinkDown** **snmp-server enable traps link cieLinkUp** **snmp-server enable traps link linkDown** **snmp-server enable traps link linkUp**
OSPF-MIB, OSPF-TRAP-MIB	**snmp-server enable traps ospf** [*tag*] **snmp-server enable traps ospf lsa** **snmp-server enable traps ospf rate-limit** *rate*
CISCO-RF-MIB	**snmp-server enable traps rf** **snmp-server enable traps rf redundancy_framework**
CISCO-RMON-MIB	**snmp-server enable traps rmon** **snmp-server enable traps rmon fallingAlarm** **snmp-server enable traps rmon hcFallingAlarm** **snmp-server enable traps rmon hcRisingAlarm** **snmp-server enable traps rmon risingAlarm**
SNMPv2-MIB	**snmp-server enable traps snmp** **snmp-server enable traps snmp authentication**
CISCO-MAC-NOTIFICATION-MIB	**snmp-server enable trap link cmn-mac-move-notification**
CISCO-PORT-STORM-CONTROL-MIB	**storm-control action trap**
CISCO-STP-EXTENSIONS-MIB	**snmp-server enable traps stpx stpxMstInconsistencyUpdate**
CISCO-STP-BRIDGE-MIB	**snmp-server enable traps bridge** **snmp-server enable traps bridge newroot** **snmp-server enable traps bridge topologychange**

MIB	Related Commands
CISCO-STPX-MIB	**snmp-server enable traps stpx** **snmp-server enable traps stpx inconsistency** **snmp-server enable traps stpx loop-inconsistency** **snmp-server enable traps stpx root-inconsistency**
CISCO-SYSTEM-EXT-MIB	**snmp-server enable traps sysmgr** **snmp-server enable traps sysmgr cseFailSwCoreNotifyExtended**
UPGRADE-MIB	**snmp-server enable traps upgrade** **snmp-server enable traps upgrade UpgradeJobStatusNotify** **snmp-server enable traps upgrade UpgradeOpNotifyOnCompletion**
VTP-MIB	**snmp-server enable traps vtp** **snmp-server enable traps vtp notifs** **snmp-server enable traps vtp vlancreate** **snmp-server enable traps vtp vlandelete**

Use the following commands in the configuration mode shown to enable the specified notification:

Command	Purpose
snmp-server enable traps **Example:** `switch(config)# snmp-server enable traps`	Enables all SNMP notifications.
snmp-server enable traps aaa [**server-state-change**] **Example:** `switch(config)# snmp-server enable traps aaa`	Enables the AAA SNMP notifications. Optionally, enables the following specific notifications: • **server-state-change**—Enables AAA server state-change notifications.
snmp-server enable traps bgp **Example:** `switch(config)# snmp-server enable traps bgp`	Enables Border Gateway Protocol (BGP) SNMP notifications.
snmp-server enable traps bridge [**newroot**] [**topologychange**] **Example:** `switch(config)# snmp-server enable traps bridge`	Enables STP bridge SNMP notifications. Optionally, enables the following specific notifications: • **newroot**—Enables STP new root bridge notifications. • **topologychange**—Enables STP bridge topology-change notifications.

Command	Purpose
snmp-server enable traps callhome [**event-notify**] [**smtp-send-fail**] **Example:** `switch(config)# snmp-server enable traps callhome`	Enables Call Home notifications. Optionally, enables the following specific notifications: • **event-notify**—Enables Call Home external event notifications. • **smtp-send-fail**—Enables Simple Mail Transfer Protocol (SMTP) message send fail notifications.
snmp-server enable traps config [**ccmCLIRunningConfigChanged**] **Example:** `switch(config)# snmp-server enable traps config`	Enables SNMP notifications for configuration changes. • **ccmCLIRunningConfigChanged**—Enables SNMP notifications for configuration changes in the running or startup configuration.
snmp-server enable traps eigrp [*tag*] **Example:** `switch(config)# snmp-server enable traps eigrp`	Enables CISCO-EIGRP-MIB SNMP notifications.
snmp-server enable traps entity [**entity_fan_status_change**] [**entity_mib_change**] [**entity_module_inserted**] [**entity_module_removed**] [**entity_module_status_change**] [**entity_power_out_change**] [**entity_power_status_change**] [**entity_unrecognised_module**] **Example:** `switch(config)# snmp-server enable traps entity`	Enables ENTITY-MIB SNMP notifications. Optionally, enables the following specific notifications: • **entity_fan_status_change**—Enables entity fan status-change notifications. • **entity_mib_change**—Enables entity MIB change notifications. • **entity_module_inserted**—Enables entity module inserted notifications. • **entity_module_removed**—Enables entity module removed notifications. • **entity_module_status_change**—Enables entity module status-change notifications. • **entity_power_out_change**—Enables entity power-out change notifications. • **entity_power_status_change**—Enables entity power status-change notifications. • **entity_unrecognised_module**—Enables entity unrecognized module notifications.

Command	Purpose
snmp-server enable traps feature-control [**FeatureOpStatusChange**] **Example:** `switch(config)# snmp-server enable traps feature-control`	Enables feature-control SNMP notifications. Optionally, enables the following specific notifications: • **FeatureOpStatusChange**—Enables feature operation status-change notifications.
snmp-server enable traps hsrp state-change **Example:** `switch(config)# snmp-server enable traps hsrp`	Enables CISCO-HSRP-MIB SNMP notifications. Optionally, enables the following specific notifications: • **state-change**—Enables HSRP state-change notifications.
snmp-server enable traps license [**notify-license-expiry**] [**notify-license-expiry-warning**] [**notify-licensefile-missing**] [**notify-no-license-for-feature**] **Example:** `switch(config)# snmp-server enable traps license`	Enables ENTITY-MIB SNMP notifications. Optionally, enables the following specific notifications: • **notify-license-expiry**—Enables license expiry notifications. • **notify-license-expiry-warning**—Enables license expiry warning notifications. • **notify-licensefile-missing**—Enables license file-missing notifications. • **notify-no-license-for-feature**—Enables no-license-installed-for-feature notifications.

Command	Purpose
snmp-server enable traps link [**cieLinkDown**] [**cieLinkUp**] [**cmn-mac-move-notification**] [**IETF-extended-linkDown**] [**IETF-extended-linkUp**] [**cisco-extended-linkDown**] [**cisco-extended-linkUp**][**linkDown**] [**linkUp**] **Example:** `switch(config)# snmp-server enable traps link`	Enables IF-MIB link notifications. Optionally, enable the following specific notifications: • **IETF-extended-linkDown**—Enables Cisco extended link state down notifications. • **IETF-extended-linkUp**—Enables Cisco extended link state up notifications. • **cmn-mac-move-notification**—Enables MAC address move notifications. • **cisco-extended-linkDown**—Enables Internet Engineering Task Force (IETF) extended link state down notifications. • **cisco-extended-linkUp**—Enables Internet Engineering Task Force (IETF) extended link state up notifications. • **linkDown**—Enables IETF link state down notifications. • **linkUp**—Enables IETF link state up notifications.
snmp-server enable traps ospf [*tag*] [**lsa**] **Example:** `switch(config)# snmp-server enable traps ospf`	Enables Open Shortest Path First (OSPF) notifications. Optionally, enables the following specific notifications: • **lsa**—Enables OSPF link state advertisement (LSA) notifications.
snmp-server enable traps rf [**redundancy-framework**] **Example:** `switch(config)# snmp-server enable traps rf`	Enables redundancy framework (RF) SNMP notifications. Optionally, enables the following specific notifications: • **redundancy-framework**—Enables RF supervisor switchover MIB notifications.

Command	Purpose
snmp-server enable traps rmon [**fallingAlarm**] [**hcFallingAlarm**] [**hcRisingAlarm**] [**risingAlarm**] **Example:** `switch(config)# snmp-server enable traps rmon`	Enables remote monitoring (RMON) SNMP notifications. Optionally, enables the following specific notifications: • **fallingAlarm**—Enables RMON falling alarm notifications. • **hcFallingAlarm**—Enables RMON high-capacity falling alarm notifications. • **hcRisingAlarm**—Enables RMON high-capacity rising alarm notifications. • **risingAlarm**—Enables RMON rising alarm notifications.
snmp-server enable traps snmp [**authentication**] **Example:** `switch(config)# snmp-server enable traps snmp`	Enables general SNMP notifications. Optionally, enables the following specific notifications: • **authentication**—Enables SNMP authentication notifications.
snmp-server enable traps stpx [**inconsistency**] [**loop-inconsistency**] [**root-inconsistency**] **Example:** `switch(config)# snmp-server enable traps stpx`	Enables SNMP STPX notifications. Optionally, enables the following specific notifications: • **inconsistency**—Enables SNMP STPX MIB inconsistency update notifications. • **loop-inconsistency**—Enables SNMP STPX MIB loop-inconsistency update notifications. • **root-inconsistency**—Enables SNMP STPX MIB root-inconsistency update notifications.
snmp-server enable traps syslog [**message-generated**] **Example:** `switch(config)# snmp-server enable traps syslog`	Sends syslog messages as traps to the defined SNMP host. Optionally, enables the following specific notifications: • **message-generated**—Enables software log message generated notifications.
snmp-server enable traps sysmgr [**cseFailSwCoreNotifyExtended**] **Example:** `switch(config)# snmp-server enable traps sysmgr`	Enables software change notifications. Optionally, enables the following specific notifications: • **cseFailSwCoreNotifyExtended**—Enables software core notifications.

Command	Purpose
snmp-server enable traps upgrade [**UpgradeJobStatusNotify**] [**UpgradeOpNotifyOnCompletion**] **Example:** `switch(config)# snmp-server enable traps upgrade`	Enables upgrade notifications. Optionally, enables the following specific notifications: • **UpgradeJobStatusNotify**—Enables upgrade job status notifications. • **UpgradeOpNotifyOnCompletion**—Enables upgrade global status notifications.
snmp-server enable traps vtp [**notifs**] [**vlancreate**] [**vlandelete**] **Example:** `switch(config)# snmp-server enable traps vtp`	Enables VTP notifications. Optionally, enables the following specific notifications: • **notifs**—Enables VTP notifications. • **vlancreate**—Enables VLAN creation notifications. • **vlandelete**—Enables VLAN deletion notifications.
storm-control action traps **Example:** `switch(config-if)# storm-control action traps`	Enables traffic storm control notifications when the traffic storm control limit is reached.

Disable Link Notifications on an Interface

You can disable linkUp and linkDown notifications on an individual interface. You can use this limit notifications on flapping interface (an interface that transitions between up and down repeatedly).

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Disable SNMP link-state traps for the interface using the command **interface***typeslot* / *port*

Example:

```
switch(config)# interface ethernet 2/2
```

This command is enabled by default.

Step 3 Disable SNMP link-state traps for the interface using the command **no snmp trap link-status**

Example:

```
switch(config-if)# no snmp trap link-status
```

This command is enabled by default.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config-if)# copy running-config startup-config
```

This is an optional step.

Display SNMP ifIndex for an Interface

The SNMP ifIndex is used across multiple SNMP MIBs to link related interface information.

Procedure

Display the persistent SNMP ifIndex value from the IF-MIB for all interfaces using the command **show interface snmp-ifindex**

Example:

```
switch# show interface snmp-ifindex | grep -i Eth12/1
 Eth12/1 441974784 (0x1a580000)
```

Optionally, use the **|** keyword and the **grep** keyword to search for a particular interface in the output.

Enable One-Time Authentication for SNMP over TCP

You can enable a one-time authentication for SNMP over a TCP session.

Procedure

Step 1 Enter the global configuration mode using the following command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Enable a one-time authentication for SNMP over a TCP session using the command **snmp-server tcp-session** [**auth**]

Example:

```
switch(config)# snmp-server tcp-session
```

The default is disabled.

Step 3 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Assign SNMP Device Contact and Location Information

You can assign the device contact information, which is limited to 32 characters (without spaces) and the device location.

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure sysContact, which is the SNMP contact name using the command **snmp-server contact***name*

Example:

```
switch(config)# snmp-server contact Admin
```

Note The *name* parameter is mandatory. However, while unconfiguring the command using the **no** form, this parameter is optional.

Step 3 Configure sysLocation, which is the SNMP location using the command **snmp-server location***name*

Example:

```
switch(config)# snmp-server location Lab-7
```

Note The *name* parameter is mandatory. However, while unconfiguring the command using the **no** form, this parameter is optional.

Step 4 Display the information about one or more destination profiles using the command **show snmp**

Example:

```
switch(config)# show snmp
```

This is an optional step.

Step 5 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure Context to Network Entity Mapping

You can configure an SNMP context to map to a logical network entity, such as a protocol instance or VRF.

Before you begin

Determine the logical network entity instance. For more information on VRFs and protocol instances, see the Cisco Nexus 9000 Series NX-OS Unicast Routing Configuration Guide or the Cisco Nexus 9000 Series NX-OS Multicast Routing Configuration Guide .

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Map an SNMP context to a protocol instance, VRF, or topology using the command [**no**] **snmp-server context***context-name* [**instance***instance-name*] [**vrf***vrf-name*] [**topology***topology-name*]

Example:

```
switch(config)# snmp-server context public1 vrf red
```

The names can be any alphanumeric string up to 32 characters. The **no** option deletes the mapping between an SNMP context and a protocol instance, VRF, or topology.

Note Do not enter an instance, VRF, or topology to delete a context mapping. If you use the instance, VRF, or topology keywords, you configure a mapping between the context and a zero-length string.

Step 3 Map an SNMPv2c community to an SNMP context using the command **snmp-server mib community-map***community-name***context***context-name*

Example:

```
switch(config)# snmp-server mib community-map public context public1
```

This is an optional step. The names can be any alphanumeric string up to 32 characters.

Step 4 Display information about one or more SNMP contexts using the command **show snmp context**

Example:

```
switch(config)# show snmp context
```

This is an optional step.

Step 5 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Disable SNMP

You can disable SNMP on the device.

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Disables SNMP using the command **no snmp-server protocol enable**

Example:

```
switch(config)# no snmp-server protocol enable
```

SNMP is enabled by default.



Note You cannot disable SNMPv1 without disabling SNMPv2. If you want to disable SNMPv1, then configure only SNMPv3, or disable SNMP entirely.

Modify SNMP Server Counter Cache Update Timer

Procedure

Step 1 Enter the global configuration mode using the command **configureterminal**

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Define how long in seconds, the port states are held in the local cache using the command **snmp-servercountercachetimeout***seconds*

Example:

```
switch(config)# snmp-server counter cache timeout 1200
```

The counter cache is enabled by default, and the default cache timeout value is 10 seconds. When disabled, the default cache timeout value is 50 seconds. The range is 1-3600.

Note For end of row (EoR) switching - The range is from 10 to 3600.

Step 3 Display the configured SNMP-server counter cache update timeout value using the command **showrunning-configsnmpall | icac**

Example:

```
switch(config)# copy running-config snmp all | i cac
```

Step 4 Disable the counter cache update using the command **nosnmp-servercountercacheenable**

Example:

```
switch(config)# no snmp-server counter cache enable
```

Note When the counter cache update is disabled, the value set in the timeout parameter determines length of time the port states are held the counter cache.

Modify AAA Synchronization Time

You can modify how long Cisco NX-OS holds the synchronized user configuration.

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure the amount of time the AAA synchronized user configuration stays in the local cache using the command **snmp-server aaa-user cache-timeout***seconds*

Example:

```
switch(config)# snmp-server aaa-user cache-timeout 1200
```

The range is from 1 to 86400 seconds. The default is 3600.

Step 3 (Optional) Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

Configure SNMP Local Engine ID

Beginning with Cisco NX-OS Release 7.0(3)I6(1), you can configure the engine ID on a local device.



Note After you configure the SNMP local engine ID, you must reconfigure all SNMP users, any host configured with the V3 users, and the community strings. Beginning with Cisco NX-OS Release 7.0(3)I7(1), you need to reconfigure only the SNMP users and community strings.

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Change the SNMP engine ID of the local device using the command **snmp-server engineID local***engineid-string*

Example:

```
switch(config)# snmp-server engineID local AA:BB:CC:1A:2C:10
```

The local engine ID should be configured as a list of colon-specified hexadecimal octets, where there are even number of hexadecimal characters that range from 10 to 64 and every two hexadecimal characters are separated by a colon. For example, 80:00:02:b8:04:61:62:63.

Step 3 Display the identification of the configured SNMP engine using the command **show snmp engineID**

Example:

```
switch(config)# show snmp engineID
```

Step 4 Disable the local engine ID and the default auto-generated engine ID is configured using the command [**no**] **snmp-server engineID local***engineid-string*

Example:

```
switch(config)# no snmp-server engineID local AA:BB:CC:1A:2C:10
```

Step 5 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

Verify SNMP Configuration

To display SNMP configuration information, perform one of the following tasks:

Command	Purpose
show interface snmp-ifindex	Displays the SNMP ifIndex value for all interfaces (from IF-MIB).
show running-config snmp [**all**]	Displays the SNMP running configuration. SNMP users brought into 10.1(1), from releases prior to 10.1(1), are displayed with the configured privacy protocol, AES-128 or DES. New users (Release 10.1(1) and later) are by default configured with AES-128 protocol. Beginning with 9.3(8) release, SNMPv3 users under show run will be represented in SALT format instead of hash.
show snmp	Displays the SNMP status.

Command	Purpose
show snmp community	Displays the SNMP community strings. **Note** If the name of the SNMP context in the **snmp-server mib community-map** command is more than 11 characters, the output of the **show snmp community** command is displayed in a vertical format instead of a tabular format.
show snmp context	Displays the SNMP context mapping.
show snmp engineID	Displays the SNMP engineID.
show snmp group	Displays SNMP roles.
show snmp host	Displays information about configured SNMP hosts.
show snmp session	Displays SNMP sessions.
show snmp source-interface	Displays information about configured source interfaces.
show snmp trap	Displays the SNMP notifications enabled or disabled.
show snmp user	Displays SNMPv3 users. Beginning with NX-OS Release 10.6(2), the display table for **SNMP USERS** contains an additional column to display the authentication mode "Auth Mode" command.

Configuration Examples

Configure Cisco NX-OS to send the Cisco linkUp or Down notifications to one notification host receiver using the Blue VRF and defines two SNMP users, Admin and NMS

```
configure terminal
snmp-server contact Admin@company.com
snmp-server user Admin auth sha abcd1234 priv abcdefgh
snmp-server user NMS auth sha abcd1234 priv abcdefgh engineID
00:00:00:63:00:01:00:22:32:15:10:03
snmp-server host 192.0.2.1 informs version 3 auth NMS
snmp-server host 192.0.2.1 use-vrf Blue
```

```
snmp-server enable traps link cisco
```

Configure SNMP to send traps using an inband port configured at the host level

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# snmp-server host 171.71.48.164 version 2c public
switch(config)# snmp-server host 171.71.48.164 source-interface ethernet 1/2
switch(config)# show snmp host
-------------------------------------------------------------------
Host Port Version Level Type SecName
-------------------------------------------------------------------
171.71.48.164 162 v2c noauth trap public
Source interface: Ethernet 1/2
-------------------------------------------------------------------
switch(config)# snmp-server host 171.71.48.164 use-vrf default
switch(config)# show snmp host
-------------------------------------------------------------------
Host Port Version Level Type SecName
-------------------------------------------------------------------
171.71.48.164 162 v2c noauth trap public
Use VRF: default
Source interface: Ethernet 1/2
-------------------------------------------------------------------
```

Configure SNMP to send traps using a globally configured inband port

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# snmp-server source-interface traps ethernet 1/2
switch(config)# show snmp source-interface
-------------------------------------------------------------------
Notification source-interface
-------------------------------------------------------------------
trap Ethernet1/2
inform -
-------------------------------------------------------------------
switch(config)# snmp-server host 171.71.48.164 use_vrf default
switch(config)# show snmp host
-------------------------------------------------------------------
Host Port Version Level Type SecName
-------------------------------------------------------------------
171.71.48.164 162 v2c noauth trap public
Use VRF: default
Source interface: Ethernet 1/2
-------------------------------------------------------------------
```

Map VRF red to the SNMPv2c public community string

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# vrf context red
switch(config-vrf)# exit
switch(config)# snmp-server context public1 vrf red
switch(config)# snmp-server mib community-map public context public1
```

Map OSPF instance Enterprise to the same SNMPv2c public community string

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# feature ospf
switch(config)# router ospf Enterprise
```

```
switch(config-router)# exit
switch(config)# snmp-server context public1 instance Enterprise
switch(config)# snmp-server mib community-map public context public1
```

SNMP Sample Outputs

SNMP EPLD Sample Output

You can view SNMP EPLD information using the following SNMP entity MIB OID'S:

1. entPhysicalName : 1.3.6.1.2.1.47.1.1.1.1.7
2. entPhysicalDescr : 1.3.6.1.2.1.47.1.1.1.1.2
3. entPhysicalContainedIn : 1.3.6.1.2.1.47.1.1.1.1.4
4. entPhysicalFirmwareRev : 1.3.6.1.2.1.47.1.1.1.1.9

Below mentioned the sample output:

```
bgl-ads-4144:167> snmpget -v 2c -c Cisco_59485 10.197.137.54 .1.3.6.1.2.1.47.1.1.1.1.2.120408
 << entPhysicalDescr
SNMPv2-SMI::mib-2.47.1.1.1.1.2.120408 = STRING: "Module-1 IO FPGA"
bgl-ads-4144:168> snmpget -v 2c -c Cisco_59485 10.197.137.54 .1.3.6.1.2.1.47.1.1.1.1.4.120408
 << entPhysicalContainedIn
SNMPv2-SMI::mib-2.47.1.1.1.1.4.120408 = INTEGER: 22
bgl-ads-4144:169> snmpget -v 2c -c Cisco_59485 10.197.137.54 .1.3.6.1.2.1.47.1.1.1.1.7.120408
 << entPhysicalName
SNMPv2-SMI::mib-2.47.1.1.1.1.7.120408 = STRING: "Module-1 IO FPGA"
bgl-ads-4144:170> snmpget -v 2c -c Cisco_59485 10.197.137.54 .1.3.6.1.2.1.47.1.1.1.1.9.120408
 << entPhysicalFirmwareRev
SNMPv2-SMI::mib-2.47.1.1.1.1.9.120408 = STRING: "0x12"
```

'Show snmp user' Sample Output

Below mentioned is the sample output for 'show snmp user' command.

```
switch# show snmp user
_____________________________________________________________________________________

                                SNMP USERS
_____________________________________________________________________________________

User                Auth      Priv(enforce) Groups              Auth Mode   acl_filter

____                ____      ____________  ______              _________   __________

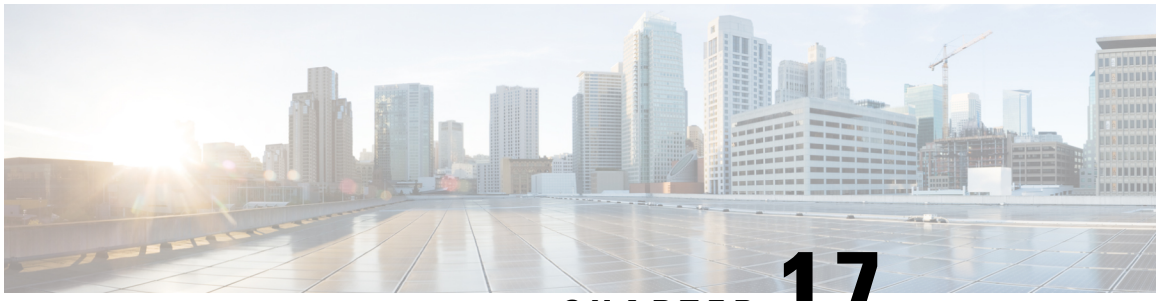
admin               md5       aes-128(no)   network-admin       Local
_____________________________________________________________________________________

                NOTIFICATION TARGET USERS (configured  for sending V3 Inform)
_____________________________________________________________________________________

User                          Auth      Priv
____                          ____      ____
```

Related Documents

Related Topic	Document Title
IP ACLs and AAA	*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*
MIBs	*Cisco Nexus 7000 Series and 9000 Series NX-OS MIB Quick Reference*
RFC 3414	*User-based Security Model (USM) for Version 3 of the Simple Network Management Protocol (SNMPv3)*
RFC 3415	*View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*
MIBs related to SNMP	To locate and download supported MIBs, go to the following URL: https://cisco.github.io/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html



CHAPTER 17

Remote Monitoring (RMON)

- Overview, on page 285
- Guidelines and Limitations, on page 287
- Configuration, on page 287
- Configure Alarms, on page 288
- Configure Events, on page 289
- Additional References, on page 290

Overview

Remote Monitoring (RMON) is a Simple Network Management Protocol (SNMP) Internet Engineering Task Force (IETF) standard monitoring specification that allows various network agents and console systems to exchange network monitoring data. Cisco NX-OS supports RMON alarms, events, and logs to monitor Cisco NX-OS devices.

An RMON alarm monitors a specific management information base (MIB) object for a specified interval, triggers an alarm at a specified threshold value (threshold), and resets the alarm at another threshold value. You can use alarms with RMON events to generate a log entry or an SNMP notification when the RMON alarm triggers.

RMON is enabled by default, but no alarms are configured in Cisco NX-OS. You can configure RMON alarms by using the CLI or an SNMP-compatible network management station.

Default Settings and Supported Features

The following table lists the default settings for RMON parameters.

Parameters	Default
RMON	Enabled
Alarms	None configured
Events	Configured (but triggered event causes nothing)

Supported Features

- Virtualization Support - Cisco NX-OS supports one instance of RMON. RMON is virtual routing and forwarding (VRF) aware. You can configure RMON to use a particular VRF to reach the RMON SMTP server.
- High Availability - Cisco NX-OS supports stateless restarts for RMON. After a reboot or supervisor switchover, Cisco NX-OS applies the running configuration.

Events

You can associate a particular event to each RMON alarm. RMON supports the following event types:

- SNMP notification—Sends an SNMP risingAlarm or fallingAlarm notification when the associated alarm triggers.
- Log—Adds an entry in the RMON log table when the associated alarm triggers.
- Both—Sends an SNMP notification and adds an entry in the RMON log table when the associated alarm triggers.

You can specify a different even for a falling alarm and a rising alarm.



Note You may choose to use the default RMON events template configuration or you can delete these entries and create new RMON events. Until you create RMON alarm configurations, no alarms will be triggered by these configurations.

Alarms

You can set an alarm on any MIB object that resolves into an SNMP INTEGER type. The specified object must be an existing SNMP MIB object in standard dot notation (for example, 1.3.6.1.2.1.2.2.1.14 represents ifInOctets.14).

When you create an alarm, you specify the following parameters:

- MIB object to monitor.
- Sampling interval—The interval that the device uses to collect a sample value of the MIB object.
- Sample type—Absolute samples take the current snapshot of the MIB object value. Delta samples take two consecutive samples and calculate the difference between them.
- Rising threshold—The value at which the device triggers a rising alarm or resets a falling alarm.
- Falling threshold—The value at which the device triggers a falling alarm or resets a rising alarm.
- Events—The action that the device takes when an alarm (rising or falling) triggers.



Note Use the hcalarms option to set an alarm on a 64-bit integer MIB object.

For example, you can set a delta type rising alarm on an error counter MIB object. If the error counter delta exceeds this value, you can trigger an event that sends an SNMP notification and logs the rising alarm event.

This rising alarm will not occur again until the delta sample for the error counter drops below the falling threshold.

Note The falling threshold must be less than the rising threshold.

Guidelines and Limitations

- You must configure an SNMP user and a notification receiver to use the SNMP notification event type.
- You can configure an RMON alarm only on a MIB object that resolves to an integer.
- When you configure an RMON alarm, the object identifier must be complete with its index so that it refers to only one object. For example, 1.3.6.1.2.1.2.2.1.14 corresponds to cpmCPUTotal5minRev, and .1 corresponds to index cpmCPUTotalIndex, which creates object identifier 1.3.6.1.2.1.2.2.1.14.1.

Configuration

Configuring RMON refers to the process of setting up Remote Monitoring (RMON) on a device.

Verify Configuration

Note The Cisco NX-OS commands for this feature may differ from those commands used in Cisco IOS.

To display RMON configuration information, perform one of the following tasks:

Command	Purpose
show rmon alarms	Displays information about RMON alarms.
show rmon events	Displays information about RMON events.
show rmon hcalarms	Displays information about RMON high-capacity alarms.
show rmon logs	Displays information about RMON logs.

Configuration Examples

This example shows how to create a delta rising alarm on ifInOctets.14 and associates a notification event with this alarm:

```
configure terminal
rmon alarm 20 1.3.6.1.2.1.2.2.1.14.1 2900 delta rising-threshold 1500 1 falling-threshold
0 owner test
rmon event 1 trap trap1
```

Configure Alarms

Before you begin

Make sure that you have configured an SNMP user and enabled SNMP notifications.

You can configure RMON alarms on any integer-based SNMP MIB object.

You can optionally specify the following parameters:

- The event number to trigger if the rising or falling threshold exceeds the specified limit.
- The owner of the alarm.

Ensure you have configured an SNMP user and enabled SNMP notifications.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Create an RMON alarm using the command **rmon alarm** *index mib-object sample-interval* { **absolute** | **delta** } **rising-threshold** *value* [*event-index*] **falling-threshold** *value* [*event-index*] [**owner** *name*]

Example:

```
switch(config)# rmon alarm 20 1.3.6.1.2.1.2.2.1.14.1 2900
delta rising-threshold 1500 1 falling-threshold 0 owner test
```

The value range is from –2147483647 to 2147483647. The owner name can be any alphanumeric string.

Step 3 Create an RMON high-capacity alarm using the command **rmon hcalarm** *index mib-object sample-interval* { **absolute** | **delta** } **rising-threshold-high** *value* **rising-threshold-low** *value* [*event-index*] **falling-threshold-high** *value* **falling-threshold-low** *value* [*event-index*] [**owner** *name*] [**storagetype** *type*]

Example:

```
switch(config)# rmon alarm 20 1.3.6.1.2.1.2.2.1.14.16777216 2900
delta rising-threshold-high 15 rising-threshold-low 151
falling-threshold-high 0 falling-threshold-low 0 owner test
```

The value range is from –2147483647 to 2147483647. The owner name can be any alphanumeric string.

The storage type range is from 1 to 5.

Step 4 Display information about RMON alarms or high-capacity alarms using the command **show rmon** { **alarms** | **hcalarms** }

Example:

```
switch(config)# show rmon alarms
```

This is an optional step.

Step 5 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure Events

- Configure RMON events to associate with RMON alarms. You can reuse the same event with multiple RMON alarms.

You can configure RMON events to associate with RMON alarms. You can reuse the same event with multiple RMON alarms.

Before you begin

Make sure you have configured an SNMP user and enabled SNMP notifications.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure an RMON event using the command **rmon event** *index* [**description** *string*] [**log**] [**trap** *string*] [**owner** *name*]

Example:

```
switch(config)# rmon event 1 trap trap1
```

The description string, trap string, and owner name can be any alphanumeric string.

Step 3 Display information about RMON events using the command **show rmon events**

Example:

```
switch(config)# show rmon events
```

This is an optional step.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

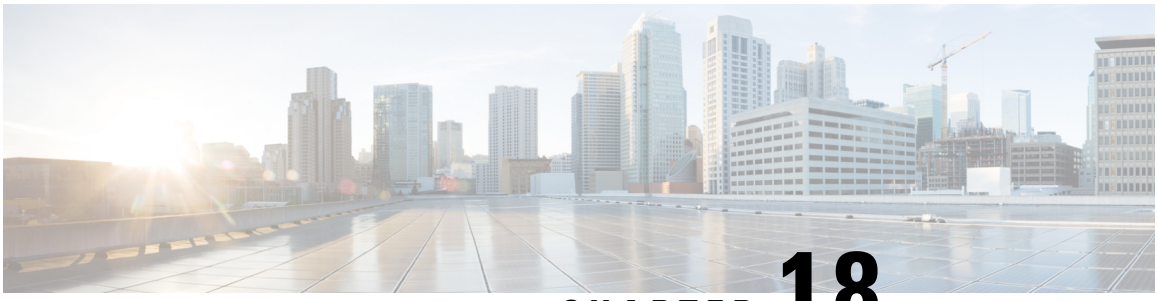
This is an optional step.

Additional References

MIBs related to RMON -

To locate and download supported MIBs, go to the following URL:

-
- https://cisco.github.io/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html



CHAPTER 18

Online Diagnostics

This chapter describes how to configure the generic online diagnostics (GOLD) feature on NX-OS devices.

- Generic Online Diagnostics, on page 291
- Guidelines and limitations for online diagnostics, on page 299
- Default settings for online diagnostics, on page 300
- Configuring Online Diagnostics, on page 300
- Commands for verifying Online Diagnostics configuration, on page 304
- Configuration examples for Online Diagnostics, on page 305

Generic Online Diagnostics

Generic Online Diagnostics (GOLD) are tests that

- run automatically during startup and continuously monitor health in the background,
- help detect hardware problems early to keep the switch running smoothly and avoid downtime, and
- check hardware and data paths while the device is running without interrupting network traffic.

Operation and types of Online Diagnostics

Online diagnostics enable you to test and verify the hardware functionality of the device while it is connected to a live network.

The online diagnostics contain tests that check different hardware components and verify the data path and control signals. Disruptive online diagnostic tests (such as the disruptive loopback test) and non-disruptive online diagnostic tests (such as the ASIC register check) run during bootup, line module online insertion and removal (OIR), and system reset. The non-disruptive online diagnostic tests run as part of the background health monitoring, and you can run these tests on demand.

Online diagnostics are categorized as bootup, runtime or health-monitoring diagnostics, and on-demand diagnostics. Bootup diagnostics run during bootup, health-monitoring tests run in the background, and on-demand diagnostics run once or at user-designated intervals when the device is connected to a live network.

Bootup diagnostics

Bootup diagnostics run during bootup and detect faulty hardware before NX-OS brings a module online. For example, if you insert a faulty module in the device, bootup diagnostics test the module and take it offline before the device uses the module to forward traffic.

Bootup diagnostics also check the connectivity between the supervisor and module hardware and the data and control paths for all the ASICs. The following table describes the bootup diagnostic tests for a module and a supervisor.

Beginning with Cisco NX-OS Release 10.6(3), the following platforms will have an additional hardware-executed Cryptographic validation before the ports are brought into service -

1. GX2A
2. GX2B
3. 9332D-H2R

Table 22: Bootup diagnostics

Diagnostic	Description
OBFL	Verifies the integrity of the onboard failure logging (OBFL) flash.
MacSecPortLoopback (Nexus 9736C-FX and 9736Q-FX line cards only)	Tests the packet path from Supervisor to each physical front panel port on the ASIC, the MACSEC capabilities of each port, and the Encryption and Decryption capabilities of the Nexus 9736C-FX and 9736Q-FX line cards. The MacSecPortLoopback test runs at boot time when the **diagnostic bootup level** is set to **complete**. The MacSecPortLoopback test runs on all the MacSec capable ports on the Nexus 9736C-FX and 9736Q-FX line cards, including ports that are broken out. The MACsec hardware is tested for the four available cipher suite algorithms: GCM-AES-128, GCM-AES-256, GCM-AES-XPN-128, and GCM-AES-XPN-256. **Note** If a MacSecPortLoopback test failure occurs, the test reports in the form of SYSLOG or OBFL. When a test failure occurs, the port is taken down and display `MACsec failure` in the **show interface** command output. You can skip the MACsec test by setting the **diagnostic bootup level** to either **minimal** or **bypass**. **Note** Beginning with Cisco NX-OS Release 10.6(3), this test is supported on N9K-C9332D-GX2B, N9K-C9348D-GX2A, N9K-C9364D-GX2A, and N9K-C9332D-H2R.
USB	Nondisruptive test. Checks the USB controller initialization on a module.
ManagementPortLoopback	Disruptive test, not an on-demand test. Tests loopback on the management port of a module.
EOBCPortLoopback	Disruptive test, not an on-demand test. Ethernet out of band.

Bootup diagnostics log failures to onboard failure logging (OBFL) and system log and trigger a diagnostic LED indication (on, off, pass, or fail).

You can configure the device to either bypass the bootup diagnostics or run the complete set of bootup diagnostics.

Runtime or health monitoring diagnostics

Runtime diagnostics, also called health monitoring (HM) diagnostics, provide information about the health of a live device and detect runtime hardware errors, memory errors, hardware degradation, software faults, and resource exhaustion.

Health monitoring diagnostics are non-disruptive and run in the background to ensure the health of a device that is processing live network traffic. You can enable or disable health monitoring tests or change their runtime interval.

The table describes the health monitoring diagnostics and test IDs for a module and a supervisor.



Note Some tests may or may not be present, depending on the capabilities of the module. A list of tests available to the module can be found using the CLI command **show diagnostic content module**<*module*>.

Table 23: Health monitoring non-disruptive diagnostics

Diagnostic	Default Interval	Default Setting	Description	Corrective Action
Module				
ACT2	30 minutes	active	Verifies the integrity of the security device on the module.	Do CallHome, log error, and disable further HM testing after 20 consecutive failures of GOLD `ACT2` test
ASICRegisterCheck	modular switches: 1 minute non-modular switches: 20 seconds and a minimum configuration default simulation interval of 10 seconds	active	Validates read/write access to the ASICs on a module.	Do CallHome, log error, and disable further HM testing for that ASIC device/instance after 20 consecutive failures of GOLD `ASICRegisterCheck` test
PrimaryBootROM	24 hours [1]	active	Verifies the integrity of the primary boot device on a module.	Do CallHome, log error, and disable further HM testing after 20 consecutive failures of GOLD `PrimaryBootROM` test

Diagnostic	Default Interval	Default Setting	Description	Corrective Action
SecondaryBootROM	24 hours [1]	active	Verifies the integrity of the secondary boot device on a module.	Do CallHome, log error, and disable further HM testing after 20 consecutive failures of GOLD `SecondaryBootROM` test
BootupPortLoopback	Only on bootup	Only on boot up - active	Checks if the supervisor to front-panel port (and back) path is operational. For every front port, the test generates a packet on an active supervisor, sends the packet toward a target port, and, using the internal loopback inside a front port, redirects the packet back to the active supervisor.	Do CallHome, Error-disable affected ports, log error testing on affected ports after 1 consecutive failures of GOLD `BootupPortLoopback` test
PortLoopback	30 minutes	active	Checks diagnostics on a per-port basis on all admin down ports.	Do CallHome, log error in Syslog/OBFL/Exception Log, and disable further HM testing on affected ports after 10 consecutive failures of GOLD `PortLoopback` test
RewriteEngineLoopback	1 minute	active	Verifies the integrity of the nondisruptive loopback for all ports up to the 1 Engine ASIC device.	Do CallHome, log error in Syslog/OBFL/Exception Log, and disable further HM testing on affected ports after 10 consecutive failures of GOLD `RewriteEngine` test
AsicMemory	Only on boot up	Only on boot up - inactive	Checks if the AsicMemory is consistent using the Mbist bit in the ASIC.	Do CallHome and log error when GOLD `AsicMemory` test fails. As the issue causing the test failure may be transient, attempt recovery reload through kernel panic. **Note** To avoid a kernel panic when the test fails, you can override the EEM system policy.

Diagnostic	Default Interval	Default Setting	Description	Corrective Action
FpgaRegTest	30 seconds	Health monitoring test - every 30 seconds - active	Test the FPGA status by read/write to FPGA.	Do CallHome, log error, disable further HM testing after 20 consecutive failures of GOLD `FpgaRegTest` test. As the issue causing the test failure may be transient, attempt recovery reload through kernel panic. **Note** To avoid a kernel panic when the test fails, you can override the EEM system policy.
L2ACLRedirect	1 minute	Health monitoring test - every minute - active	Checks if the active inband path is operational. The test generates a packet on an active supervisor through the active fabric module. It then sends the packet toward the front panel port (physical interface on the line card) and, using the ACL entry, redirects the packet back to the active supervisor.	Do CallHome, log error, disable further HM testing after 10 consecutive failures of L2ACLRedirect test. As the issue causing the test failure may be transient, attempt recovery reload through kernel panic. **Note** To avoid a kernel panic when the test fails, you can override the EEM system policy.
OBFL	30 minutes	active	Verifies the integrity of the onboard failure logging (OBFL) flash, and monitors for available storage in the device.	

Diagnostic	Default Interval	Default Setting	Description	Corrective Action
FabricConnectivityTest	1 minute	active	Verifies fabric/linecard link status. Validates that the fabric links are functioning. **Note** Only available on Nexus 9500-R series line cards and N9K-X9836DM-A line cards. This is also supported on X98366DM-A and X98900CD-A line cards with Nexus 9808 and 9804 switches.	
FabricReachabilityTest	1 minute	active	Verifies fabric/linecard reachability status. Validates that each fabric component has a valid path to every other fabric component in the system. **Note** Only available on Nexus 9500-R series line cards and N9K-X9836DM-A line cards. This is also supported on X98366DM-A and X98900CD-A line cards with Nexus 9808 and 9804 switches.	
Supervisor				
Backplane	30 minutes	active	Verifies the integrity of the backplane SPROM devices.	

Diagnostic	Default Interval	Default Setting	Description	Corrective Action
NVRAM	5 minutes	active	Verifies the sanity of the NVRAM blocks on a supervisor.	Do CallHome, log error, and disable further HM testing after 20 consecutive failures of GOLD `NVRAM` test
RealTimeClock	5 minutes	active	Verifies that the real-time clock on the supervisor is ticking.	Do CallHome, log error, and disable further HM testing after 20 consecutive failures of GOLD `RealTimeClock` test
PrimaryBootROM	30 minutes	active	Verifies the integrity of the primary boot device on the supervisor.	Do CallHome, log error, and disable further HM testing after 20 consecutive failures of GOLD `PrimaryBootROM` test
SecondaryBootROM	30 minutes	active	Verifies the integrity of the secondary boot device on the supervisor.	Do CallHome, log error, and disable further HM testing after 20 consecutive failures of GOLD `SecondaryBootROM` test
BootFlash	30 minutes	active	Verifies access to the bootflash devices.	Do CallHome and log error when GOLD `BootFlash` test fails
USB	30 minutes	active	Verifies access to the USB devices.	Do Call Home and log error when GOLD `USB` test fails
SystemMgmtBus	30 seconds	active	Verifies the availability of the system management bus.	Do Call Home, log error, and disable further HM testing for that fan or power supply after 20 consecutive failures of GOLD `SystemMgmtBus` test
Mce	30 minutes	Health monitoring test - 30 minutes - active	This test uses the mcd_dameon and reports any machine check error reported by the Kernel.	Do CallHome, log error, and disable further HM testing after 20 consecutive failures of GOLD `Mce` test
Pcie	Only on boot up	Only on boot up - inactive	Reads PCIe status registers and check for any error on the PCIe device.	Do CallHome and log error when GOLD `Pcie` test fails
Console	Only on boot up	Only on boot up - inactive	This runs a port loopback test on the management port on boot up to check for its consistency.	Do CallHome, log error, and disable further HM testing after 20 consecutive failures of GOLD `Console` test

Diagnostic	Default Interval	Default Setting	Description	Corrective Action
FpgaRegTest	30 seconds	Health monitoring test - every 30 seconds - active	Test the FPGA status by read/write to FPGA. **Note** FpgaRegTest on Nexus 9808 and 9804 switches for Fabric Modules (19-26) will be displayed under the FpgaRegTest result of Active-SUP.	Do CallHome, log error, disable further HM testing after 20 consecutive failures of GOLD `FpgaRegTest` test. As the issue causing the test failure may be transient, attempt recovery reload through kernel panic. **Note** To avoid a kernel panic when the test fails, you can override the EEM system policy.

[1] Minimum configurable test interval is 6 hours

On-demand diagnostics

On-demand tests help localize faults and are usually needed in one of these situations:

- To respond to an event that has occurred, such as isolating a fault.
- In anticipation of an event that may occur, such as a resource exceeding its utilization limit.

You can run all the health monitoring tests on demand. You can schedule on-demand diagnostics to run immediately.

You can also modify the default interval for a health monitoring test.

High Availability

A key part of high availability is detecting hardware failures and taking corrective action while the device runs in a live network.

Online diagnostics in high availability detect hardware failures and provide feedback to high availability software components to make switchover decisions.

NX-OS supports stateless restarts for online diagnostics. After a reboot or supervisor switchover, NX-OS applies the running configuration.

Virtualization support

Online diagnostics are virtual routing and forwarding (VRF) aware. You can configure online diagnostics to use a particular VRF to reach the online diagnostics SMTP server.

Guidelines and limitations for online diagnostics

This topic describes the guidelines and limitations for Generic Online Diagnostics (GOLD) configuration.

- The Nexus platform switches that do not support the BootupPortLoopback test on breakout ports are:
  - • Nexus N9K-C9808, Nexus N9K-C9804
    - Nexus 9364E-SG2-Q, Nexus 9364E-SG2-O
    - Nexus N9K-X9836DM-A
    - Nexus N9K-C9232E-B1
    - Nexus 9336C-SE1
    - N9396Y12C-SE1, N9396T12C-SE1
    - N9324C-SE1U, N9348Y2C6D-SE1U

  On these platforms, the BootupPortLoopback diagnostic test is not supported on breakout ports during bootup. The test may display as Untested (U) or Fail (F) for breakout subports. On-demand PortLoopback tests are supported on breakout ports after bootup.

- You cannot run disruptive online diagnostic tests on demand.
- Interface Rx and Tx packet counters are incremented (approximately four packets every 15 minutes) for ports in the shutdown state.
- The PortLoopback test is periodic, so the packet counter is incremented on admin down ports every 30 minutes. The test runs only on admin down ports. When a port is unshut, the counters are not affected.
- When a port fails for the per-port BootupPortLoopback test, the port enters the error-disabled state. (To remove this state, enter the **shutdown** and **no shutdown** commands on the port.)
- Beginning with NX-OS Release 10.6(1)F, on Nexus modular switches, if the Backplane diagnostic test fails and a BACKPLANE_AUTHENTICATION_FAIL syslog appears, do not perform an upgrade or a system reload.

Platform support

- The Nexus platform switches and line cards that do not support the run-time PortLoopback test but support the BootupPortLoopback test:

  Switches

  Line Cards

  - Nexus 9736C-EX
  - Nexus 97160YC-EX
  - Nexus 9732C-EX
  - Nexus 9732C-EXM

- Beginning with NX-OS Release 10.3(1)F, Generic Online Diagnostics (GOLD) is supported on the Nexus 9800 platform switches.
- Beginning with NX-OS Release 10.4(1)F, Generic Online Diagnostics (GOLD) is supported on the following line cards and switches:
  - Nexus 9804 switch
  - Nexus C9332D-H2R switch
  - Nexus X98900CD-A line card
  - Nexus X98900CD-A and X9836DM-A line cards with Nexus 9808 and 9804 switches
- Beginning with NX-OS Release 10.4(2)F, Generic Online Diagnostics (GOLD) is supported on Nexus 93400LD-H1 platform switches.
- Beginning with NX-OS Release 10.4(3)F, Generic Online Diagnostics (GOLD) is supported on Nexus 9364C-H1 platform switch.
- Beginning with NX-OS Release 10.5(3)F, Generic Online Diagnostics (GOLD) is supported on N9364E-SG2-O and N9364E-SG2-Q switches. However, BootupPortLoopback test is not supported on any breakout ports on these switches.
- Beginning with NX-OS Release 10.6(2)F, Generic On-line Diagnostics (GOLD) is supported on N9396Y12C-SE1 and N9396T12C-SE1 switches.
- Beginning with NX-OS Release 10.6(2n), Generic On-line Diagnostics (GOLD) is supported on N9164E-NS4-O switch. However, the NVRAM test is not supported on this switch.

Default settings for online diagnostics

This table lists the default settings for online diagnostic parameters.

Parameters	Default
`Bootup diagnostics level`	complete
`Nondisruptive tests`	active

Configuring Online Diagnostics

Note Be aware that the NX-OS commands for this feature may differ from those commands used in IOS.

Set the bootup diagnostic level

You can configure the bootup diagnostics to run the complete set of tests, or you can bypass all bootup diagnostic tests for a faster module bootup time.

Note We recommend that you set the bootup online diagnostics level to `complete`. We do not recommend bypassing the bootup online diagnostics.

Procedure

Step 1 Enter global configuration mode using the **configure terminal** command.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Configure the bootup diagnostic level to trigger diagnostics when the device boots using the **diagnostic bootup level** {**complete** | | **bypass**} command.

Example:

```
switch(config)# diagnostic bootup level complete
```

The bootup diagnostic levels include:

- **complete**—Perform a complete set of bootup diagnostics. The default is complete.
- **bypass**—Do not perform any bootup diagnostics.

Step 3 (Optional) Display the bootup diagnostic level (bypass or complete) that is currently in place on the device using the **show diagnostic bootup level** command.

Example:

```
switch(config)# show diagnostic bootup level
```

Step 4 (Optional) Copy the running configuration to the startup configuration using the **copy running-config startup-config** command.

Example:

```
switch(config)# copy running-config startup-config
```

Activate a diagnostic test

You can set a diagnostic test as active and optionally modify the interval (in hours, minutes, and seconds) at which the test runs.

Procedure

Step 1 **configure terminal**

Example:

```
switch# configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 Configure the interval at which the specified test is run using the **diagnostic monitor interval module** *slot* **test** [*test-id* | *name* | **all**] **hour** *hour* **min** *minute* **second** *second* command.

Example:

```
switch(config)# diagnostic monitor interval module 6 test 3 hour 1 min 0 second 0
```

If no interval is set, the test runs at the interval set previously, or the default interval.

The argument ranges are as follows:

- *slot*—The range is from 1 to 10.
- *test-id*—The range is from 1 to 14.
- *name*—Can be any case-sensitive, alphanumeric string up to 32 characters.
- *hour*—The range is from 0 to 23 hours.
- *minute*—The range is from 0 to 59 minutes.
- *second*—The range is from 0 to 59 seconds.

Step 3 Activate the specified test using the [**no**] **diagnostic monitor module** *slot* **test** [*test-id* | *name* | **all**] command.

Example:

```
switch(config)# diagnostic monitor interval module 6 test 3
```

The argument ranges are:

- *slot*—The range is from 1 to 10.
- *test-id*—The range is from 1 to 14.
- *name*—Can be any case-sensitive, alphanumeric string up to 32 characters.

The [**no**] form of this command inactivates the specified test. Inactive tests keep their current configuration but do not run at the scheduled interval.

Step 4 (Optional) Display information about the diagnostics and their attributes using the **show diagnostic content module** {*slot* | **all**} command.

Example:

```
switch(config)# show diagnostic content module 6
```

Start or stop an on-demand diagnostic test

You can start or stop an on-demand diagnostic test. You can optionally modify the number of iterations to repeat this test, and the action to take if the test fails.

We recommend that you only manually start a disruptive diagnostic test during a scheduled network maintenance time.

Procedure

Step 1 (Optional) Configure the number of times that the on-demand test runs using the **diagnostic ondemand iteration** *number* command.

The range is from 1 to 999. The default is 1.

Example:

```
switch# diagnostic ondemand iteration 5
```

Step 2 (Optional) Configure the action to take if the on-demand test fails using the **diagnostic ondemand action-on-failure** {**continue failure-count** *num-fails* | **stop**} command.

The *num-fails* range is from 1 to 999. The default is 1.

Example:

```
switch# diagnostic ondemand action-on-failure stop
```

Step 3 Required: Start one or more diagnostic tests on a module using the **diagnostic start module** *slot* **test** [*test-id* | *name* | **all** | **non-disruptive**] [**port** *port-number* | **all**] command.

The module slot range is from 1 to 10. The *test-id* range is from 1 to 14. The test name can be any case-sensitive, alphanumeric string up to 32 characters. The port range is from 1 to 48.

Example:

```
switch# diagnostic start module 6 test all
```

Step 4 Required: Stop one or more diagnostic tests on a module using the **diagnostic stop module** *slot* **test** [*test-id* | *name* | **all**] command.

The module slot range is from 1 to 10. The *test-id* range is from 1 to 14. The test name can be any case-sensitive, alphanumeric string up to 32 characters.

Example:

```
switch# diagnostic stop module 6 test all
```

Step 5 (Optional) Verify that the diagnostic has been scheduled using the **show diagnostic status module** *slot* command.

Example:

```
switch# show diagnostic status module 6
```

Simulate diagnostic results

You can simulate a diagnostic test result.

Procedure

Simulates a test result using the **diagnostic test simulation module** *slot* **test** *test-id* {**fail** | **random-fail** | **success**} [**port** *number* | **all**] command.

The *test-id* range is from 1 to 14. The port range is from 1 to 48.

Example:

```
switch# diagnostic test simulation module 2 test 2 fail
```

Clear diagnostic results

You can clear diagnostic test results.

Procedure

Step 1 Clear the test result for the specified test using the **diagnostic clear result module** [*slot* | **all**] **test** {*test-id* | **all**} command.

The argument ranges are:

- *slot*—The range is from 1 to 10.
- *test-id*—The range is from 1 to 14.

Example:

```
switch# diagnostic clear result module 2 test all
```

Step 2 Clear the simulated test result using the **diagnostic test simulation module** *slot* **test** *test-id* clear command.

The *test-id* range is from 1 to 14.

Example:

```
switch# diagnostic test simulation module 2 test 2 clear
```

Commands for verifying Online Diagnostics configuration

Use any of the show commands provided in the table to display the required configuration information about online diagnostics.

Command	Purpose
show diagnostic bootup level	Displays information about bootup diagnostics.
show diagnostic content module {*slot* | **all**}	Displays information about diagnostic test content for a module.
show diagnostic description module *slot* **test** [*test-name* | **all**]	Displays the diagnostic description.
show diagnostic events [**error** | **info**]	Displays diagnostic events by error and information event type.
show diagnostic ondemand setting	Displays information about on-demand diagnostics.

Command	Purpose
show diagnostic result module *slot* [**test** [*test-name* | **all**]] [**detail**]	Displays information about the results of a diagnostic.
show diagnostic simulation module *slot*	Displays information about a simulated diagnostic.
show diagnostic status module *slot*	Displays the test status for all tests on a module.
show hardware capacity [**eobc** | **forwarding** | **interface** | **module** | **power**]	Displays information about the hardware capabilities and current hardware utilization by the system.
show module	Displays module information including the online diagnostic test status.

Configuration examples for Online Diagnostics

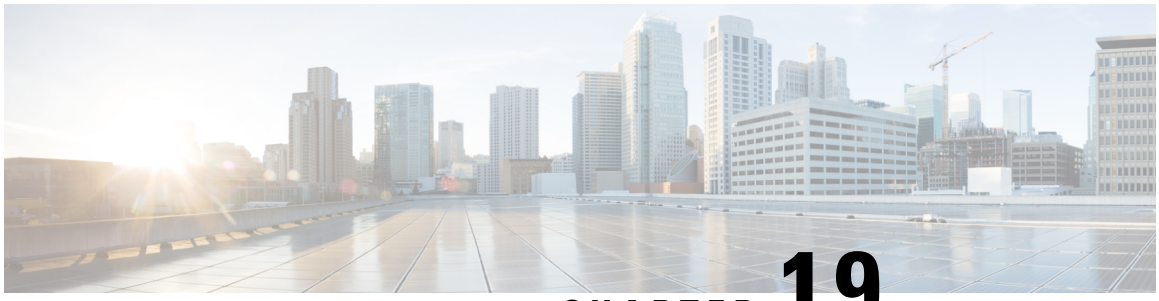
This topic provides configuration examples for starting and monitoring online diagnostics on modules.

This example shows how to start all on-demand tests on module 6.

```
diagnostic start module 6 test all
```

This example shows how to activate test 2 and set the test interval on module 6.

```
configure terminal
diagnostic monitor module 6 test 2
diagnostic monitor interval module 6 test 2 hour 3 min 30 sec 0
```



CHAPTER 19

Configuring the Embedded Event Manager

This chapter describes how to configure the Embedded Event Manager (EEM) to detect and handle critical events on Cisco NX-OS devices.

- Embedded Event Manager, on page 307
- Prerequisites for Embedded Event Manager, on page 311
- Guidelines and limitations for Embedded Event Manager, on page 311
- Default settings for EEM, on page 314
- Configure Embedded Event Manager, on page 314
- Commands for verifying the Embedded Event Manager configuration, on page 327
- Configuration examples for Embedded Event Manager, on page 327
- Event Log Auto-Collection and backup, on page 329

Embedded Event Manager

An Embedded Event Manager is an automation feature that

- monitors device events and triggers configured actions,
- includes event statements, action statements, and policies, and
- automates troubleshooting, recovery, and notifications.

Embedded Event Manager (EEM) is a software component that monitors device events and takes automated actions to recover or troubleshoot, as configured by the user.

EEM consists of three major components:

- Event statements—Events to monitor from another NX-OS component that may require some action, workaround, or notification.
- Action statements—An action that EEM can take, such as executing command line interface commands, sending an email through the use of Smart Call Home feature, and disabling an interface to recover from an event.
- Policies—An event that is paired with one or more actions to troubleshoot or recover from the event.

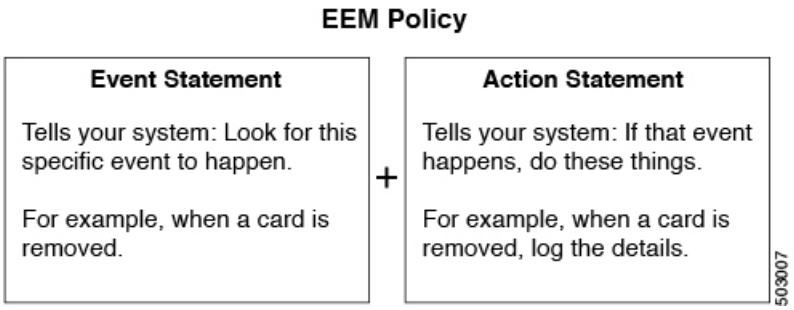
Policies

An Embedded Event Manager (EEM) policy consists of an event statement and one or more action statements.

The event statement defines the event to look for as well as the filtering characteristics for the event. The action statement defines the action EEM takes when the event occurs.

This figure shows the two basic statements in an EEM policy.

Figure 6: EEM Policy Statements



You can configure EEM policies using the command-line interface or a VSH script.

EEM gives you a device-wide view of policy management. You configure EEM policies on the supervisor, and EEM pushes the policy to the correct module based on the event type. EEM takes any actions for a triggered event either locally on the module or on the supervisor (the default option).

EEM maintains event logs on the supervisor.

NX-OS has a number of preconfigured system policies. These system policies define many common events and actions for the device. System policy names begin with two underscore characters (__).

You can create user policies to suit your network. If you create a user policy, any actions in your policy occur after EEM triggers any system policy actions that are related to the same event as your policy.

You can also override some system policies. The overrides that you configure take the place of the system policy. You can override the event or the actions.

Use the **show event manager system-policy** command to view the preconfigured system policies and determine which policies that you can override.

Note You should use the **show running-config eem** command to check the configuration of each policy. An override policy that consists of an event statement and no action statement triggers no action and no notification of failures.

Note Your override policy should always include an event statement. An override policy without an event statement overrides all possible events in the system policy.

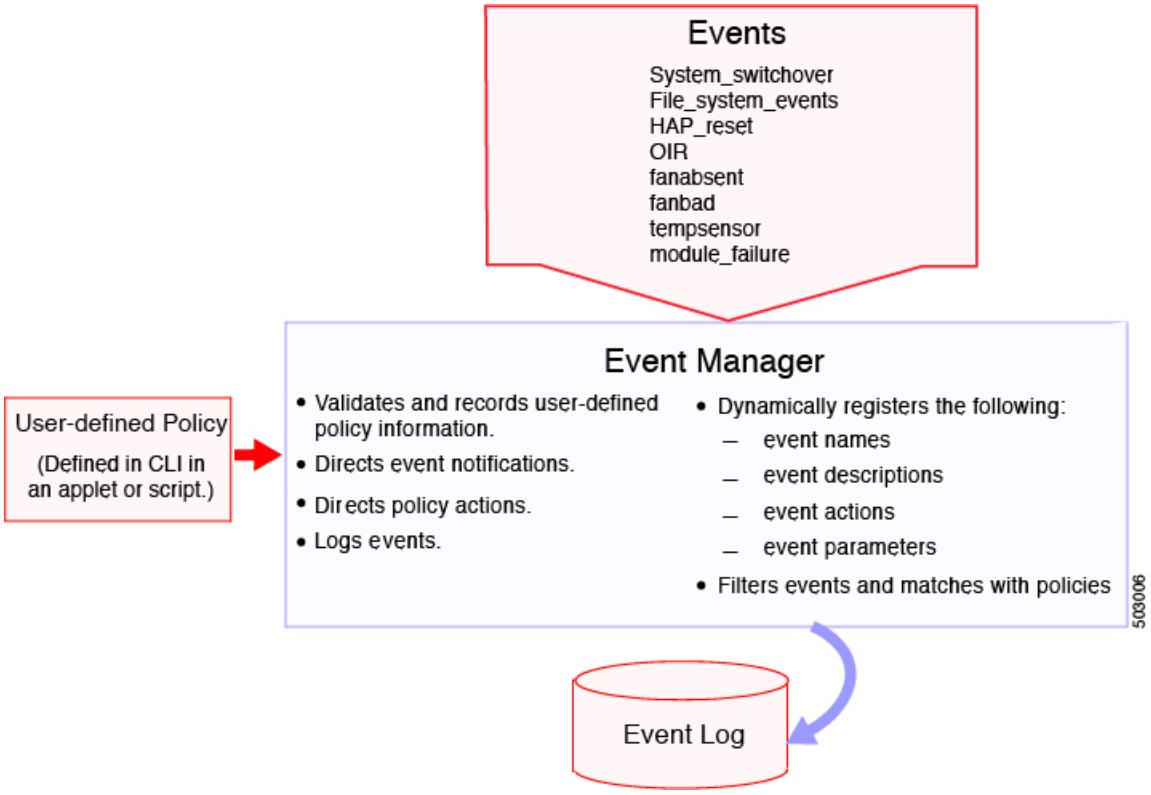
Event statements

Event statements specify the event that triggers a policy to run.

Embedded Event Manager (EEM) defines event filters so only critical events or multiple occurrences of an event within a specified time period trigger an associated action.

This figure shows events that are handled by EEM.

Figure 7: EEM Overview



You can configure multiple event triggers. EEM schedules and runs policies on the basis of event statements. EEM examines the event and action commands and runs them as defined.

Note If you want to allow the triggered event to process any default actions, you must configure the EEM policy to allow the event default action statement.

Action statements

Action statements describe the action triggered by a policy. Each policy can have multiple action statements. If no action is associated with a policy, Embedded Event Manager (EEM) still observes events but takes no actions.

EEM supports these actions in action statements:

- Device and module operations
  - Execute any command line interface commands
  - Force the shutdown of any module
  - Reload the device
  - Shut down specified modules because the power is over budget
- Monitoring and notification
  - Log an exception
  - Generate a syslog message
  - Generate a Call Home event
  - Generate an SNMP notification
- System and policy handling
  - Update a counter
  - Use the default action for the system policy



Note EEM can only process a complete action command line interface list of up to 1024 characters in total. If more actions are required, you must define them as a new redundant applet with same trigger.



Note If you want to allow the triggered event to process any default actions, you must configure the EEM policy to allow the default action. For example, if you match a command line interface command in a match statement, you must add the event-default action statement to the EEM policy or EEM does not allow the command line interface command to execute.



Note Verify that your action statements within your user policy or overriding policy do not negate each other or adversely affect the associated system policy.

VSH script policies

You can also write policies in a VSH script, using a text editor. These policies have an event statement and action statement(s) just as other policies, and these policies can either augment or override system policies. After you write your VSH script policy, copy it to the device and activate it.

Environment variables

You can define environment variables for EEM that are available for all policies.

Environment variables are useful for configuring common values that you can use in multiple policies. For example, you can create an environment variable for the IP address of an external email server.

You can use an environment variable in action statements by using the parameter substitution format.

This example shows a sample action statement to force a module 1 shutdown, with a reset reason of `"EEM action"`.

```
switch (config-eem-policy)# action 1.0 forceshut module 1 reset-reason “EEM action.”
```

If you define an environment variable for the shutdown reason, called default-reason, you can replace that reset reason with the environment variable, as shown in the following example.

```
switch (config-eem-policy)# action 1.0 foreshut module 1 reset-reason $default-reason
```

You can reuse this environment variable in any policy.

EEM event correlation

You can trigger an EEM policy based on a combination of events.

First, you use the **tag** keyword to create and differentiate multiple events in the EEM policy. Then using a set of boolean operators (**and**, **or**, **andnot**), along with the count and time, you can define a combination of these events to trigger a custom action.

High Availability

NX-OS supports stateless restarts for EEM. After a reboot or supervisor switchover, NX-OS applies the running configuration.

Virtualization support

Not all actions or events are visible. You must have network-admin privileges to configure policies.

Prerequisites for Embedded Event Manager

The prerequisites Embedded Event Manager (EEM) include:

- You must have network-admin user privileges to configure EEM.

Guidelines and limitations for Embedded Event Manager

This topic describes the configuration guidelines and limitations for Embedded Event Manager (EEM).

General

- NX-OS does not support evmc restart.

- Action with option **collect** must be always first action in the event applet statement.
- Only 10 triggers from the same client (for example, vshd is the client for `event cli`, snmp is the client for `event snmp` and so on) are allowed to be published within one second.
- You can invoke EEM from Python. For more information about Python, see the Cisco Nexus 9000 Series NX-OS Programmability Guide.

Policies

- The maximum number of configurable EEM policies is 500.
- Action statements within your user policy or overriding policy should not negate each other or adversely affect the associated system policy.
- To allow a triggered event to process any default actions, you must configure the EEM policy to allow the default action. For example, if you match a command in a match statement, you must add the event-default action statement to the EEM policy or EEM will not allow the command to execute.
- When more than one event statement is included in an EEM policy, each event statement must have a **tag** keyword with a unique tag argument.
- Default action execution is not supported for policies that are configured with tagged events.
- Note these statements about override policies:
  - An override policy that consists of an event statement without an action statement triggers no action and no notification of failures.
  - An override policy without an event statement overrides all possible events in the system policy.

Commands

- When you configure an EEM policy action to collect **show tech** commands, make sure to allocate enough time for the **show tech** commands to complete before the same action is called again.
- The following rules apply to regular command expressions:
  - All regular expressions must conform to the Portable Operating System Interface for uniX (POSIX) extended standard.
  - All keywords must be expanded.
  - Only the * symbol can be used for argument replacement.

EEM event correlation

- EEM event correlation is supported only on the supervisor module.
- EEM event correlation is not supported across different modules within a single policy.
- EEM event correlation supports up to four event statements in a single policy. The event types can be the same or different, but only these event types are supported: cli, counter, module, module-failure, oir, snmp, and syslog.
- EEM event correlation does not override the system default policies.

Event log collection and backup

- The guidelines that apply to Event Log Auto-Collection and Backup are:
  - By default, enabled log collection on a switch provides between 15 minutes to several hours of event logs depending on size, scale and component activity.
  - To be able to collect relevant logs that span a longer period, only enable event log retention for the specific services/features you need. See *Enabling Extended Log File Retention For a Single Service*. You can also export the internal event logs. See *External Log File Storage*.
  - When troubleshooting, it is good practice to manually collect a snapshot of internal event logs in real time. See *Generating a Local Copy of Recent Log Files*.

Release-specific

- Beginning with NX-OS Release 10.3(1)F, the default auto-collect is not supported with system switchover. On system switchover, re-run the bloggerd auto-collect commands on the new Active supervisor to enable auto-collect for respective components.
- Beginning with NX-OS Release 10.3(3)F, default bloggerd auto-collect is supported for adjmgr, cts, l2fm, and vmtracker.
- Beginning with NX-OS Release 10.4(1)F, default bloggerd auto-collect is supported for additional components ipqosmgr, aclqos, cfs, ethport, feature-mgr, icam, interface manager, lacp, m2rib, mfdm, nbm, ngoam, nve, port-channel, qos, sla_responder, sla_sender, sla_twamp, smm, spm, sysmgr, and vpc.
  - The minimum configurable purge time is increased from 0 to 48 hours.
  - Files that cross 14 days are purged automatically, regardless of the reserved bootflash space (a maximum of 5%).
  - When the maximum reserved space for auto-collect is in use, new auto-collections are rejected until space becomes available again in the reserved space after file purging or manual file deletion.
- Beginning with NX-OS Release 10.4(3)F, you can use event policy-default disable command to disable system policies from an applet overriding the system policy. You need to apply this command to the applet overriding the system policy to stop them from triggering. If you have multiple applets overriding the same system policy, with some of them have event policy-default disable is configured, those applets will not be triggered but other applets which are configured with a different event override will still trigger the system policy.
  - When a system policy is overridden multiple times with the same event, only the last policy is executed.
  - Also in the above case, if there is one policy hitting in less time, only that will get executed not other policies.
- Beginning with NX-OS Release 10.5(3)F, until a line card is fully operational, any custom overriding policies of system-policies related to the line card are not triggered. Only the default system-policies are honored. The logs for any triggers of line card-related system-policies are only available after the line card is fully brought up.

Default settings for EEM

This table lists the default settings for EEM parameters.

Parameters	Default
System policies	Active

Configure Embedded Event Manager

You can create policies that contain actions to take based on system policies.

To display information about the system policies, use the **show event manager system-policy** command.

Defining an Environment Variable

You can define a variable to serve as a parameter in an EEM policy.

SUMMARY STEPS

1. **configure terminal**
2. **event manager environment** *variable-name variable-value*
3. (Optional) **show event manager environment** {*variable-name* | **all**}
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

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

Step 2	**event manager environment** *variable-name variable-value* **Example:** `switch(config)# event manager environment emailto "admin@anyplace.com"`	Creates an environment variable for EEM. The *variable-name* can be any case-sensitive, alphanumeric string up to 29 characters. The *variable-value* can be any quoted alphanumeric string up to 39 characters.
Step 3	(Optional) **show event manager environment** {*variable-name* | **all**} **Example:** `switch(config)# show event manager environment all`	Displays information about the configured environment variables.

	Command or Action	Purpose
Step 4	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Define a user policy using the command line interface

You can define a user policy using the command line interface to the device.

Procedure

Step 1 Enter global configuration mode using the **configure terminal** command.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Register the applet with EEM and enters applet configuration mode using the **event manager applet** *applet-name* command.

The *applet-name* can be any case-sensitive, alphanumeric string up to 29 characters.

Example:

```
switch(config)# event manager applet monitorShutdown
switch(config-applet)#
```

Step 3 (Optional) Configure a descriptive string for the policy using the **description** *policy-description* command.

The string can be any alphanumeric string up to 80 characters. Enclose the string in quotation marks.

Example:

```
switch(config-applet)# description "Monitors interface shutdown."
```

Step 4 Configure the event statement for the policy using the **event** *event-statement* command.

Example:

```
switch(config-applet)# event cli match "conf t ; interface * ; shutdown"
```

Repeat this step for multiple event statements. See Configure event statements, on page 316.

Step 5 (Optional) Correlate multiple events in the policy using the **tag** *tag* {**and** | **andnot** | **or**} *tag* [**and** | **andnot** | **or** {*tag*}] {**happens** *occurs in seconds*} command.

The range for the *occurs* argument is from 1 to 4294967295. The range for the *seconds* argument is from 0 to 4294967295 seconds.

Example:

```
switch(config-applet)# tag one or two happens 1 in 10000
```

Step 6 Configure an action statement for the policy using the **action** *number*[*.number2*] *action-statement* command.

Example:

```
switch(config-applet)#action 1.0 cli show interface Ethernet 3/1
```

Repeat this step for multiple action statements. See Configure action statements, on page 320.

Step 7 (Optional) Display information about the status of the configured policy using the **show event manager policy-state** *name* [**module** *module-id*] command.

Example:

```
switch(config-applet)# show event manager policy-state monitorShutdown
```

Step 8 (Optional) Copy the running configuration to the startup configuration using the **copy running-config startup-config** command.

Example:

```
switch(config)# copy running-config startup-config
```

Configure event statements

Use one of these commands in applet configuration mode to configure an event statement.

Command	Purpose
event application [**tag** *tag*] **sub-system** *sub-system-id* **type** *event-type* Example: `switch(config-applet)# event application sub-system 798 type 1`	Triggers an event when an event specification matches the subsystem ID and application event type. The range for the *sub-system-id* and for the *event-type* is from 1 to 4294967295. The **tag** *tag* keyword-argument pair identifies this specific event when multiple events are included in the policy. **Note** To use this command, you must first enable the **feature evmed** command to enable generic event detectors.
event cli [**tag** *tag*] **match** *expression* [**count** *repeats* | **time** *seconds*] Example: `switch(config-applet)# event cli match "conf t ; interface * ; shutdown"`	Triggers an event if you enter a command that matches the regular expression. The **tag** *tag* keyword-argument pair identifies this specific event when multiple events are included in the policy. The *repeats* range is from 1 to 65000. The time range, in seconds, is from 0 to 4294967295, where 0 indicates no time limit.

Command	Purpose
event counter [**tag** *tag*] **name** *counter* **entry-val** *entry* **entry-op** {**eq** | **ge** | **gt** | **le** | **lt** | **ne**} [**exit-val** *exit* **exit-op** {**eq** | **ge** | **gt** | **le** | **lt** | **ne**}] Example: `switch(config-applet)# event counter name mycounter entry-val 20 gt`	Triggers an event if the counter crosses the entry threshold based on the entry operation. The event resets immediately. Optionally, you can configure the event to reset after the counter passes the exit threshold. The **tag** *tag* keyword-argument pair identifies this specific event when multiple events are included in the policy. The *counter* name can be any case-sensitive, alphanumeric string up to 28 characters. The *entry* and *exit* value ranges are from 0 to 2147483647.
event fanabsent [**fan** *number*] **time** *seconds* Example: `switch(config-applet)# event fanabsent time 300`	Triggers an event if a fan is removed from the device for more than the configured time, in seconds. The *number* range is module-dependent. The *seconds* range is from 10 to 64000.
event fanbad [**fan** *number*] **time** *seconds* Example: `switch(config-applet)# event fanbad time 3000`	Triggers an event if a fan fails for more than the configured time, in seconds. The *number* range is module-dependent. The *seconds* range is from 10 to 64000.
event fib {**adjacency extra** | **resource tcam usage** | **route** {**extra** | **inconsistent** | **missing**}} Example: `switch(config-applet)# event fib adjacency extra`	Triggers an event for one of the following: • **adjacency extra**—If there is an extra route in the unicast FIB. • **resource tcam usage**—Each time the TCAM utilization percentage becomes a multiple of 5, in either direction. • **route** {**extra** | **inconsistent** | **missing**}—If a route is added, changed, or deleted in the unicast FIB.
event gold module {*slot* | **all**} **test** *test-name* [**severity** {**major** | **minor** | **moderate**}] **testing-type** {**bootup** | **monitoring** | **ondemand** | **scheduled**} **consecutive-failure** *count* Example: `switch(config-applet)# event gold module 2 test ASICRegisterCheck testing-type ondemand consecutive-failure 2`	Triggers an event if the named online diagnostic test experiences the configured failure severity for the configured number of consecutive failures. The *slot* range is from 1 to 10. The *test-name* is the name of a configured online diagnostic test. The *count* range is from 1 to 1000.
event memory {**critical** | **minor** | **severe**} Example: `switch(config-applet)# event memory critical`	Triggers an event if a memory threshold is crossed. See also Configure memory thresholds, on page 324.

Command	Purpose
event module [**tag** *tag*] **status** {**online** | **offline** | **any**} **module** {**all** | *module-num*} Example: `switch(config-applet)# event module status offline module all`	Triggers an event if the specified module enters the selected status. The **tag** *tag* keyword-argument pair identifies this specific event when multiple events are included in the policy.
event module-failure [**tag** *tag*] **type** *failure-type* **module** {*slot* | **all**} **count** *repeats* [**time** *seconds*] Example: `switch(config-applet)# event module-failure type lc-failed module 3 count 1`	Triggers an event if a module experiences the failure type configured. The **tag** *tag* keyword-argument pair identifies this specific event when multiple events are included in the policy. The *repeats* range is from 0 to 4294967295. The *seconds* range is from 0 to 4294967295, where 0 indicates no time limit.
event none Example: `switch(config-applet)# event none`	Manually runs the policy event without any events specified. **Note** To use this command, you must first enable the **feature evmed** command to enable generic event detectors.
event oir [**tag** *tag*] {**fan** | **module** | **powersupply**} {**anyoir** | **insert** | **remove**} [*number*] Example: `switch(config-applet)# event oir fan remove 4`	Triggers an event if the configured device element (fan, module, or power supply) is inserted or removed from the device. The **tag** *tag* keyword-argument pair identifies this specific event when multiple events are included in the policy. You can optionally configure a specific fan, module, or power supply number. The *number* range is as follows: • Fan number—Module dependent. • Module number—Device dependent. • Power supply number—The range is from 1 to 3.
event policy-default count *repeats* [**time** *seconds*] Example: `switch(config-applet)# event policy-default count 3`	Uses the event configured in the system policy. Use this option for overriding policies. The *repeats* range is from 1 to 65000. The *seconds* range is from 0 to 4294967295, where 0 indicates no time limit.

Command	Purpose
event poweroverbudget Example: `switch(config-applet)# event poweroverbudget`	Triggers an event if the power budget exceeds the capacity of the configured power supplies.
event snmp [**tag** *tag*] **oid** *oid* **get-type** {**exact** | **next**} **entry-op** {**eq** | **ge** | **gt** | **le** | **lt** | **ne**} **entry-val** *entry* [**exit-comb** {**and** | **or**}] **exit-op** {**eq** | **ge** | **gt** | **le** | **lt** | **ne**} **exit-val** *exit* **exit-time** *time* **polling-interval** *interval* Example: `switch(config-applet)# event snmp oid 1.3.6.1.2.1.31.1.1.1.6 get-type next entry-op lt 300 entry-val 0 exit-op eq 400 exit-time 30 polling-interval 300`	Triggers an event if the SNMP OID crosses the entry threshold based on the entry operation. The event resets immediately, or optionally you can configure the event to reset after the counter passes the exit threshold. The OID is in dotted decimal notation. The **tag** *tag* keyword-argument pair identifies this specific event when multiple events are included in the policy. The *entry* and *exit* value ranges are from 0 to 18446744073709551615. The time, in seconds, is from 0 to 2147483647. The interval, in seconds, is from 1 to 2147483647.
event storm-control Example: `switch(config-applet)# event storm-control`	Triggers an event if traffic on a port exceeds the configured storm control threshold.
event syslog [**occurs** *count*] {**pattern** *string* | **period** *time* | **priority** *level* | **tag** *tag*} Example: `switch(config-applet)# event syslog period 500`	Triggers an event if the specified syslog threshold is exceeded. The range for the count is from 1 to 65000, and the range for the time is from 1 to 4294967295. The priority range is from 0 to 7. The **tag** *tag* keyword-argument pair identifies this specific event when multiple events are included in the policy.
event sysmgr memory [**module** *module-num*] **major** *major-percent* **minor** *minor-percent* **clear** *clear-percent* Example: `switch(config-applet)# event sysmgr memory minor 80`	Triggers an event if the specified system manager memory threshold is exceeded. The range for the percentage is from 1 to 99.
event sysmgr switchover count *count* **time** *interval* Example: `switch(config-applet)# event sysmgr switchover count 10 time 1000`	Triggers an event if the specified switchover count is exceeded within the time interval specified. The switchover count is from 1 to 65000. The time interval is from 0 to 2147483647.

Command	Purpose
event temperature [**module** *slot*] [*sensor-number*] **threshold** {**any** | **major** | **minor**} Example: `switch(config-applet)# event temperature module 2 threshold any`	Triggers an event if the temperature sensor exceeds the configured threshold. The sensor range is from 1 to 18.
event timer {**absolute time** *time* **name** *name* | **countdown time** *time* **name** *name* | **cron cronentry** *string* | **tag** *tag* | **watchdog time** *time* **name** *name*} Example: `switch(config-applet)# event timer absolute time 100 name abtimer`	Triggers an event if the specified time is reached. The range for the time is from 1 to 4294967295. • **absolute time**—Triggers an event when the specified absolute time of day occurs. • **countdown time**—Triggers an event when when the specified time counts down to zero. The timer does not reset. • **cron cronentry**—Triggers an event when the cron string specification matches the current time. • **watchdog time**—Triggers an event when the specified time counts down to zero. The timer automatically resets to the initial value and continues to count down. The **tag** *tag* keyword-argument pair identifies this specific event when multiple events are included in the policy. **Note** To use this command, you must first enable the **feature evmed** command to enable generic event detectors.
event track [**tag** *tag*] *object-number* **state** {**any** | **down** | **up**} Example: `switch(config-applet)# event track 1 state down`	Triggers an event if the tracked object is in the configured state. The **tag** *tag* keyword-argument pair identifies this specific event when multiple events are included in the policy. The *object-number* range is from 1 to 500.

Configure action statements

Use the commands in the table in EEM configuration mode to configure the required action statements.

Command	Purpose
action *number*[*.number2*] **cli** *command1* [*command2...*] [**local**] Example: `switch(config-applet)# action 1.0 cli show interface Ethernet 3/1`	Runs the configured CLI commands. You can optionally run the commands on the module where the event occurred. The action label is in the format *number1.number2*. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9.
action *number*[*.number2*] **counter name** *counter* **value** *val* **op** {**dec** | **inc** | **nop** | **set**} Example: `switch(config-applet)# action 2.0 counter name mycounter value 20 op inc`	Modifies the counter by the configured value and operation. The action label is in the format *number1.number2*. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9. The counter name can be any case-sensitive, alphanumeric string up to 28 characters. The *val* can be an integer from 0 to 2147483647 or a substituted parameter.
action *number*[*.number2*] **event-default** Example: `switch(config-applet)# action 1.0 event-default`	Executes the default action for the associated event. The action label is in the format *number1.number2*. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9.
action *number*[*.number2*] **forceshut** [**module** *slot* | **xbar** *xbar-number*] **reset-reason** *seconds* Example: `switch(config-applet)# action 1.0 forceshut module 2 reset-reason "flapping links"`	Forces a module, crossbar, or the entire system to shut down. The action label is in the format *number1.number2*. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9. The reset reason is a quoted alphanumeric string up to 80 characters.
action *number*[*.number2*] **overbudgetshut** [**module** *slot*[*-slot*]] Example: `switch(config-applet)# action 1.0 overbudgetshut module 3-5`	Forces one or more modules or the entire system to shut down because of a power overbudget issue. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9.
action *number*[*.number2*] **policy-default** Example: `switch(config-applet)# action 1.0 policy-default`	Executes the default action for the policy that you are overriding. The action label is in the format *number1.number2*. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9.
action *number*[*.number2*] **publish-event** Example: `switch(config-applet)# action 1.0 publish-event`	Forces the publication of an application-specific event. The action label is in the format *number1.number2*. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9.

Command	Purpose
action *number*[*.number2*] **reload** [**module** *slot*[*-slot*]] Example: `switch(config-applet)# action 1.0 reload module 3-5`	Forces one or more modules or the entire system to reload. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9.
action *number*[*.number2*] **snmp-trap** {[**intdata1** *data* [**intdata2** *data*]] [**strdata** *string*]} Example: `switch(config-applet)# action 1.0 snmp-trap strdata "temperature problem"`	Sends an SNMP trap with the configured data. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9. The *data* arguments can by any number up to 80 digits. The *string* can be any alphanumeric string up to 80 characters.
action *number*[*.number2*] **syslog** [**priority** *prio-val*] **msg** *error-message* Example: `switch(config-applet)# action 1.0 syslog priority notifications msg "cpu high"`	Sends a customized syslog message at the configured priority. *number* can be any number up to 16 digits. The range for *number2* is from 0 to 9. The *error-message* can be any quoted alphanumeric string up to 80 characters.

Note If you want to allow the triggered event to process any default actions, you must configure the EEM policy to allow the default action. For example, if you match a command line interface command in a match statement, you must add the event-default action statement to the EEM policy or EEM will not allow the command line interface command to execute. You can use the **terminal event-manager bypass** command to allow all EEM policies with command line interface matches to execute the command line interface command.

Define a policy using a VSH script

You can define a policy using a VSH script.

Before you begin

Ensure that you are logged in with administrator privileges.

Ensure that your script name is the same name as the script filename.

Procedure

Step 1 In a text editor, list the commands that define the policy.

Step 2 Name the text file and save it.

Step 3 Copy the file to the following system directory: bootflash://eem/user_script_policies.

Register and activate a VSH script policy

You can register and activate a policy defined in a VSH script.

Procedure

Step 1 Enter global configuration mode using the **configure terminal** command.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Register and activate an EEM script policy using the **event manager policy** *policy-script* command.

The *policy-script* can be any case-sensitive alphanumeric string up to 29 characters.

Example:

```
switch(config)# event manager policy moduleScript
```

Step 3 (Optional) Copy the running configuration to the startup configuration using the **copy running-config startup-config** command.

Example:

```
switch(config)# copy running-config startup-config
```

Override a policy

You can override a system policy.

Procedure

Step 1 Enter global configuration mode using the **configure terminal** command.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 (Optional) Display information about the system policy that you want to override, including thresholds, using the **show event manager policy-state** *system-policy* command.

Use the **show event manager system-policy** command to find the system policy names. For information about system policies, see Embedded Event Manager System Events and Configuration Examples, on page 659.

Example:

```
switch(config-applet)# show event manager policy-state __ethpm_link_flap
Policy __ethpm_link_flap
Cfg count : 5
Cfg time interval : 10.000000 (seconds)
Hash default, Count 0
```

Step 3 Override a system policy and enter applet configuration mode using the **event manager applet** *applet-name* **override** *system-policy* command.

The *applet-name* can be any case-sensitive alphanumeric string up to 29 characters. The *system-policy* must be one of the existing system policies.

Example:

```
switch(config)# event manager applet ethport override __ethpm_link_flap
switch(config-applet)#
```

Step 4 (Optional) Configure a descriptive string for the policy using the **description** *policy-description* command.

The string can be any alphanumeric string up to 80 characters. Enclose the string in quotation marks.

Example:

```
description "Overrides link flap policy."
```

Step 5 Required: Configure or disable the event statement for the policy using the **[no] event** {*event-statement* | *policy-default disable*} command.

Example:

```
switch(config-applet)# event policy-default count 2 time 1000
```

The **no** form of this command removes the configuration.

Step 6 Required: Configure an action statement for the policy using the **action** *number action-statement* command.

Example:

```
switch(config-applet)# action 1.0 syslog priority warnings msg "Link is flapping."
```

Repeat this step for multiple action statements.

Step 7 (Optional) Display information about the configured policy using the **show event manager policy-state** *name* command.

Example:

```
switch(config-applet)# show event manager policy-state ethport
```

Step 8 (Optional) Copy the running configuration to the startup configuration using the **copy running-config startup-config** command.

Example:

```
switch(config)# copy running-config startup-config
```

Configure memory thresholds

You can set the memory thresholds that are used to trigger events and set whether the operating system should kill processes if it cannot allocate memory.

Before you begin

Ensure that you are logged in with administrator privileges.

Procedure

Step 1 **configure terminal** command.

Example:

```
switch# configure terminal
switch(config)#
```

Enters global configuration mode.

Step 2 Configure the system memory thresholds that generate EEM memory events using the **system memory-thresholds minor** *minor* **severe** *severe* **critical** *critical* command.

Example:

```
switch(config)# system memory-thresholds minor 60 severe 70 critical 80
```

The default values are as follows:

- Minor-85
- Severe-90
- Critical-95

When the memory thresholds are exceeded, the system generates syslogs.

- 2013 May 7 17:06:30 switch %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : MINOR
- 2013 May 7 17:06:30 switch %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : SEVERE
- 2013 May 7 17:06:30 switch %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : CRITICAL
- 2013 May 7 17:06:35 switch %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : MINOR ALERT RECOVERED
- 2013 May 7 17:06:35 switch %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : SEVERE ALERT RECOVERED
- 2013 May 7 17:06:35 switch %$ %PLATFORM-2-MEMORY_ALERT: Memory Status Alert : CRITICAL ALERT RECOVERED

Step 3 (Optional) Configure the system to not kill processes when the memory cannot be allocated using the **system memory-thresholds threshold critical no-process-kill** command.

The default value is to allow the system to kill processes, starting with the one that consumes the most memory.

Example:

```
switch(config)# system memory-thresholds threshold critical no-process-kill
```

Step 4 (Optional) Display information about the system memory configuration using the **show running-config | include "system memory"** command.

Example:

```
switch(config-applet)# show running-config | include "system memory"
```

Step 5 (Optional) Copy the running configuration to the startup configuration using the **copy running-config startup-config** command.

Example:

```
switch(config)# copy running-config startup-config
```

Configure syslog as EEM publisher

Before you begin

EEM should be available for registration by syslog.

The syslog daemon must be configured and executed.

You can monitor syslog messages from the switch.



Note The maximum number of searchable strings to monitor syslog messages is 10.

Procedure

Step 1 Enter global configuration mode using the **configure terminal** command.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Register an applet with EEM and enter applet configuration mode using the **event manager applet** *applet-name* command.

Example:

```
switch(config)# event manager applet abc
switch(config-applet)#
```

Step 3 Monitor syslog messages and invokes the policy based on the search string in the policy using the **event syslog** [**tag** *tag*] {**occurs** *number* | **period** *seconds* | **pattern** *msg-text* | **priority** *priority*} command.

- The **tag** *tag* keyword-argument pair identifies this specific event when multiple events are included in the policy.
- The **occurs** *number* keyword-argument pair specifies the number of occurrences. The range is from 1 to 65000.
- The **period** *seconds* keyword-argument pair specifies the interval during which the event occurs. The range is from 1 to 4294967295.
- The **pattern** *msg-text* keyword-argument pair specifies the matching regular expression. The pattern can contain character text, an environment variable, or a combination of the two. If the string contains embedded blanks, it is enclosed in quotation marks.
- The **priority** *priority* keyword-argument pair specifies the priority of the syslog messages. If this keyword is not selected, all syslog messages are set at the informational priority level.

Example:

```
switch(config-applet)# event syslog occurs 10
```

Step 4 (Optional) Copy the running configuration to the startup configuration using the **copy running-config startup-config** command.

Example:

```
switch(config)# copy running-config startup-config
```

Commands for verifying the Embedded Event Manager configuration

Perform any of the show commands to view the required EEM configuration information.

Command	Purpose
show event manager environment [*variable-name* | **all**]	Displays information about the event manager environment variables.
show event manager event-types [*event* | **all** | **module** *slot*]	Displays information about the event manager event types.
show event manager history events [**detail**] [**maximum** *num-events*] [**severity** {**catastrophic** | **minor** | **moderate** | **severe**}]	Displays the history of events for all policies.
show event manager policy-state *policy-name*	Displays information about the policy state, including thresholds.
show event manager script system [*policy-name* | **all**]	Displays information about the script policies.
show event manager system-policy [**all**]	Displays information about the predefined system policies.
show running-config eem	Displays information about the running configuration for EEM.
show startup-config eem	Displays information about the startup configuration for EEM.

Configuration examples for Embedded Event Manager

This topic provides configuration examples for Embedded Event Manager (EEM) policies, including how to override system policies, send syslog messages, trigger SNMP notifications, and correlate multiple events.

This example shows how to override the __lcm_module_failure system policy by changing the threshold for just module 3 hitless upgrade failures. This example also sends a syslog message. The settings in the system policy, __lcm_module_failure, apply in all other cases.

```
event manager applet example2 override __lcm_module_failure
event module-failure type hitless-upgrade-failure module 3 count 2
action 1 syslog priority errors msg module 3 “upgrade is not a hitless upgrade!”
action 2 policy-default
```

This example shows how to override the __ethpm_link_flap system policy and shuts down the interface:

```
event manager applet ethport override __ethpm_link_flap
event policy-default count 2 time 1000
action 1 cli conf t
action 2 cli int et1/1
action 3 cli no shut
```

This example creates an EEM policy that allows the CLI command to execute but triggers an SNMP notification when a user enters configuration mode on the device:

```
event manager applet TEST
event cli match "conf t"
action 1.0 snmp-trap strdata "Configuration change"
action 2.0 event-default
```

Note You must add the **event-default** action statement to the EEM policy or EEM will not allow the CLI command to execute.

This example shows how to correlate multiple events in an EEM policy and execute the policy based on a combination of the event triggers. In this example, the EEM policy is triggered if one of the specified syslog patterns occurs within 120 seconds.

```
event manager applet eem-correlate
event syslog tag one pattern "copy bootflash:.* running-config.*”
event syslog tag two pattern “copy run start”
event syslog tag three pattern “hello”
tag one or two or three happens 1 in 120
action 1.0 reload module 1
```

Upon reaching a maximum failure threshold, the AsicMemory, FpgaRegTest, and L2ACLRedirect system policies force a reload of the switch. This example shows how to override the default action for one of these policies and issue a syslog instead:

```
event manager applet gold override __fpgareg
action 1 syslog priority emergencies msg FpgaRegTest_override
```

This example shows how to override a default policy but still enact the default action:

```
event manager applet gold_fpga_ovrd override __fpgareg
  action 1 policy-default
  action 2 syslog priority emergencies msg FpgaRegTest_override
```

Note For additional EEM configuration examples, see Embedded Event Manager System Events and Configuration Examples, on page 659.

Event Log Auto-Collection and backup

Automatically collected event logs are stored locally on switch memory. Event log file storage is a temporary buffer that stores files for a fixed amount of time. Once the time period has elapsed, a roll-over of the buffer makes room for the next files. The roll-over uses a first-in-first-out method.

EEM uses these methods of collection and backup:

- Extended Log File Retention
- Trigger-Based Event Log Auto-Collection

Extended Log File Retention

All Nexus platform switches, with at least 8Gb of system memory, support the extended retention of event logging files. Storing the log files locally on the switch or remotely through an external container, reduces the loss of event logs due to rollover.

Enable Extended Log File Retention for all services

Extended Log File Retention is enabled by default for all services running on a switch. If the switch does not have the log file retention feature enabled (**no bloggerd log-dump** is configured), use this procedure to enable it.

Procedure

Step 1 Enter global configuration mode using the **configure terminal** command.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Enable the log file retention feature for all services using the **bloggerd log-dump all** command.

Example:

```
switch(config)# bloggerd log-dump all
switch(config)#
```

```
switch# configure terminal
switch(config)# bloggerd log-dump all
Sending Enable Request to Bloggerd
Bloggerd Log Dump Successfully enabled
switch(config)#
```

Disable Extended Log File Retention for all services

Extended Log File Retention is enabled by default for all services running on the switch.

If the switch has the log file retention feature enabled for all services, and you want to disable it, use this procedure.

Procedure

Step 1 Enter global configuration mode using the **configure terminal** command.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Disable the log file retention feature for all services on the switch using the **no bloggerd log-dump all** command.

Example:

```
switch(config)# no bloggerd log-dump all
switch(config)#
```

```
switch# configure terminal
switch(config)# no bloggerd log-dump all
Sending Disable Request to Bloggerd
Bloggerd Log Dump Successfully disabled
switch(config)#
```

Enable Extended Log File Retention for a single service

Use this procedure to enable Log File Retention for a single service.

Procedure

Step 1 Display information about the ACL Manager including the service SAP number using the **show system internal sysmgr service name***service-type* command.

Example:

```
switch# show system internal sysmgr service name aclmgr
```

Step 2 Enter global configuration mode using the **configure terminal** command.

Example:

```
switch# configure terminal
switch(config)#
```

Step 3 Enable the log file retention feature for the ACL Manager service using the **bloggerd log-dump sap***number* command.

Example:

```
switch(config)# bloggerd log-dump sap 351
```

Step 4 Display information about the log file retention feature on the switch using the **show system internal bloggerd info log-dump-info** command.

Example:

```
switch(config)# show system internal bloggerd info log-dump-info
```

```
switch# show system internal sysmgr
service name aclmgr Service "aclmgr" ("aclmgr", 80):
UUID = 0x182, PID = 653, SAP = 351
State: SRV_STATE_HANDSHAKED (entered at time Mon Nov  4 11:10:41 2019).
Restart count: 1
Time of last restart: Mon Nov  4 11:10:39 2019.
The service never crashed since the last reboot.
Tag = N/A
Plugin ID: 0
switch(config)# configure terminal
switch(config)# bloggerd log-dump sap 351
Sending Enable Request to Bloggerd
Bloggerd Log Dump Successfully enabled
switch(config)# show system internal bloggerd info log-dump-info
------------------------------------------------------------
Log Dump config is READY
Log Dump is DISABLED for ALL application services in the switch
Exceptions to the above rule (if any) are as follows:
------------------------------------------------------------
Module     | VDC        | SAP                       | Enabled?
------------------------------------------------------------
1          | 1          | 351 (MTS_SAP_ACLMGR     ) | Enabled
------------------------------------------------------------
------------------------------------------------------------
Log Dump Throttle Switch-Wide Config:
-------------------------------------
Log Dump Throttle                                    : ENABLED
Minimum buffer rollover count (before throttling)    : 5
Maximum allowed rollover count per minute            : 1
------------------------------------------------------------

switch(config)#
```

Display Extended Log Files

Use this task to display the event log files currently stored on the switch.

Procedure

Display the event log files currently stored on the switch using the **dir debug:log-dump/** command.

Example:

```
switch# dir debug:log-dump/
```

```
switch# dir debug:log-dump/

3676160 Dec 05 02:43:01 2019 20191205023755_evtlog_archive.tar
3553280 Dec 05 06:05:06 2019 20191205060005_evtlog_archive.tar

Usage for debug://sup-local
913408 bytes used
```

```
4329472 bytes free
5242880 bytes total
```

Display global dictionary per log statistics

This commands displays the statistics of log message being logged by each component with a counter, to store the number of times a log being repeated from the system up time.

Procedure

Display the per log statistics of each component using the **show system internal sdwrap buffers sap** <*sap-num*> **dict-stats detailed** command.

Example:

```
switch# show system internal sdwrap buffers sap <sap-num> dict-stats detailed
```

```
switch# show system internal sdwrap buffers sap 221 dict-stats detailed

Sap received is: 221

SDWrap Format Strings Dictionary stats for sap MTS_SAP_L2FM (221)

UUID: SRVUUID_LIBSDWRAP, Inst Type: 0

MsgId Frequency Message                                                 --------- ---------
 ------------------------------
 4    1 System is not undergoing ISSU
78    1 Vlan %d is part of reserved vlan bmp from sdb                       179   1 Vlan
%d is not found in L2FM database. Skipping the delete request 306   1 Vlan %d is removed
from L2FM database and MTM database
416   1 mts_drap_get_my_local_swid_only_msg failed with rc %#x
496   1 Lookup for backplane mac failed for vdc %d with st = %s
598   1 L2FM - Slot %d SwCardId %d Port %d - %d Fp %d Cli %d
```

Disable Extended Log File Retention for a single service

Extended Log File Retention is enabled by default for all services on the switch. If the switch has the log file retention feature enabled for a single service or all services, and you want to disable a specific service or services, use this procedure.

Procedure

Step 1 Display information about the Access Control List (ACL) Manager including the service Service Access Point (SAP) number using the **show system internal sysmgr service name***service-type* command.

Example:

```
switch# show system internal sysmgr service name aclmgr
```

Step 2 Enter global configuration mode using the **configure terminal** command.

Example:

```
switch# configure terminal
switch(config)#
```

Step 3 Disable the log file retention feature for the ACL Manager service using the **no bloggerd log-dump sap***number* command.

Example:

```
switch(config)# no bloggerd log-dump sap 351
```

Step 4 Display information about the log file retention feature on the switch using the **show system internal bloggerd info log-dump-info** command.

Example:

```
switch(config)# show system internal bloggerd info log-dump-info
```

This example shows how to disable extended log file retention for a service named `aclmgr`.

```
switch# show system internal sysmgr service name aclmgr
Service "aclmgr" ("aclmgr", 80):
        UUID = 0x182, PID = 653, SAP = 351
        State: SRV_STATE_HANDSHAKED (entered at time Mon Nov  4 11:10:41 2019).
        Restart count: 1
        Time of last restart: Mon Nov  4 11:10:39 2019.
        The service never crashed since the last reboot.
        Tag = N/A
        Plugin ID: 0
switch(config)# configure terminal
switch(config)# no bloggerd log-dump sap 351
Sending Disable Request to Bloggerd
Bloggerd Log Dump Successfully disabled
switch(config)# show system internal bloggerd info log-dump-info
------------------------------------------------------------
Log Dump config is READY
Log Dump is DISABLED for ALL application services in the switch
Exceptions to the above rule (if any) are as follows:
------------------------------------------------------------
Module     | VDC        | SAP                       | Enabled?
------------------------------------------------------------
1          | 1          | 351 (MTS_SAP_ACLMGR     ) | Disabled
------------------------------------------------------------
------------------------------------------------------------
Log Dump Throttle Switch-Wide Config:
-------------------------------------
Log Dump Throttle                                  : ENABLED
Minimum buffer rollover count (before throttling)      : 5
Maximum allowed rollover count per minute              : 1
------------------------------------------------------------

switch(config)#
```

Trigger-based event log auto-collection

CTWG Nov 2025

Automatically collect relevant data when issues occur using trigger-based log collection.

- No impact on control plane.

- Customizable configuration:
  - Defaults populated by Cisco.
  - Selectively override what-to-collect by network administrator or by Cisco TAC.
  - Automatically update new triggers on image upgrades.
- Store logs locally on the switch or remotely on an external server.
- Supports severity 0, 1, and 2 syslogs.
- Supports unexpected protocol events such as BGP, BFD, OSPF, and ISIS.
- Custom syslogs for ad-hoc events (auto-collection commands attached to the syslogs).

Enable trigger-based log file auto-collection

To enable trigger-based automatic creation of log files, you must create an override policy for the `__syslog_trigger_default` system policy with a custom YAML file and define the specific logs for which information is collected.

For more information on creating a custom YAML file to enable log file auto-collection, see Auto-Collection YAML file contents, on page 340.

Log-profile YAML files

A log-profile YAML file is a configuration file that

- sets logging parameters,
- limits log exports per minute and log file rotations to manage storage,
- and is located at `/bootflash/log_profile.yaml`.

The local backup storage, designated for backing up the event history of various components, is a shared resource. To prevent any single component from consuming the entire storage capacity, two key parameters are set within the bloggerd log profile: **rollovers_allowed** and **rotations_allowed**.

The log-profile YAML file, named as `log_profile.yaml`, specifies these bloggerd extended logging parameters. It controls the number of exports per minute (**rollovers_allowed**) and the retention of log files (**rotations_allowed**) for each component, essentially dictating how much space each component is allowed to use within the backup storage memory.

It is generally advisable not to alter the default values unless directed by the support engineers.

The parameters that can be adjusted for each component in the `Log-Profile` file are:

- **rollovers_allowed**—This parameter determines the maximum number of times data can be exported from a component to the backup storage memory within a minute.
- **rotations_allowed**—This parameter sets the number of log file rotations (or versions) that are retained in the backup storage before the system overwrites them, effectively limiting the buffer space used by each component's logs.

To reflect the changes made in `/bootflash/log_profile.yaml` file, execute the **bloggerd reparse log-profile** command during run time at bloggerd.

This is an example of a default `log_profile.yaml` file which is packaged as part of the image. The definitions for the keys/values in the file are mentioned in the table.

Example log-profile YAML file

This is an example of a default `log_profile.yaml` file which is packaged as part of the image.

```
273:
    entry_1:
          srv_uuid: 273
          instance: 0
          rollovers_allowed: 250
          rotations_allowed: 5
          mod: sup

274:
    entry_1:
          srv_uuid: 274
          instance: 0
          rollovers_allowed: 250
          rotations_allowed: 5
          mod: sup
```

Table 24: Log-profile YAML Key/Value definitions

Key: Value	Description
273	UUID of the component whose sdwrap buffer throttling needs to be overridden.
entry_1:	Only one entry supported per component. Up to 20 entries can be made per component. Each entry is identified **`entry_1`** through **`entry_20`**.
srv_uuid:	Each sdwrap log buffer is identified with (uuid, instance id) tuple.
instance:	Sdwrap log buffer instance id with respect to srv_uuid field above. A **`"-1"`** means all instances.
rollovers_allowed:	How many rollovers allowed per minute. **`0-500`** allowed value.
rotations_allowed:	How many rotations allowed per throttle.
mod:	Name of the syslog component (`platform` is a facility name in syslog).

Note It is advisable not to alter the default values unless directed by the support engineers.

Auto-Collection YAML files

The Auto-Collection YAML file is a file that

- is specified in the **action** command in the EEM function,
- defines actions for different system or feature components, and
- specifies what data should be collected automatically during trigger-based event log auto-collection.

Auto-Collection YAML file usage and precedence

This file is located in the switch directory: /bootflash/scripts. In addition to the default YAML file, you can create component-specific YAML files and place them in the same directory. The naming convention for component-specific YAML files is **component-name.yaml**. If a component-specific file is present in the same directory, it takes precedence over the file that is specified in the **action** command. For example, if the action file, **bootflash/scripts/platform.yaml** is in the **/bootflash/scripts** directory with the default action file, **bootflash/scripts/test.yaml**, then the instructions defined in **platform.yaml** file take precedence over the instructions for the platform component present in the default **test.yaml** file.

Examples of components are Address Resolution Protocol (ARP), Border Gateway Protocol (BGP), Intermediate System to Intermediate System (IS-IS), and so on. If you are not familiar with all the component names, contact Customer Support for assistance in defining the YAML file for component-specific actions (and for the default **test.yaml** file as well).

Example for Auto-Collection YAML file in EEM applet

```
event manager applet test_1 override __syslog_trigger_default
  action 1.0 collect test.yaml $_syslog_msg
```

Create or delete Auto-Collection per component

Beginning with NX-OS Release 10.2(2)F, the auto-collect adoption improvement feature allows you to control the auto-collection for a single or set of components based on your requirement.

Note Beginning with NX-OS Release 10.3(1)F, multiple components are enabled by default and the YAML file of the component is copied to the default auto-collect folder. However, you can disable and enable bloggerd auto-collect component using this command.

Note The YAML file is editatble and handle it with caution. If the file gets corrupted with any syntax, tar will not be generated. .

You can use this command for creation or deletion of auto-collect YAML files.

```
switch# bloggerd auto-collect component <component_name> {enable | disable}
```

Beginning with NX-OS Release 10.4(3)F, auto-collect is enabled by default on ufdm, ipfib, mrib, pim, eltm, and iftmc components. You can view the list of default enabled auto-collect components using the **dir bootflash:scripts/default-autocollect** command.

To disable the required components, use **bloggerd auto-collect component <component_name> {enable | disable}** command.

The collected logs are saved in `bootflash:eem_snapshots` folder `dir bootflash:eem_snapshots`.

When you use the enable command, the YAML file of the component is copied from the backup folder to the default auto-collect folder. Note that you cannot copy the contents of the backup-staging folder as it is a read-only folder; whereas, you can copy the contents of the default auto-collect folder (`bootflash:scripts` folder), if required.

When you use the disable command, the YAML file of the component is removed from the default auto-collect folder under the `bootflash:scripts` folder.

Caution As the yaml is editable, you need use this file with caution. If the file is corrupted with any syntax issues, the tarball will not be generated.

YAML files are located in:

```
Spine_1# run bash sudo su
 bash-4.4# ls /bootflash/scripts/default-autocollect/
```

YAML files:

```
/bootflash/scripts/default-autocollect/m6rib.yaml
/bootflash/scripts/default-autocollect/mrib.yaml
/bootflash/scripts/default-autocollect/pim6.yaml
/bootflash/scripts/default-autocollect/pim.yaml
```

Note You must manually clear the `bootfalsh:eem_snapshots` folder. The logs are not collected otherwise.

To view the syslogs of auto collect in the following scenarios by enabling the logging like below:

1. if auto collect fails
2. tarball collection

Logging enablement:

```
logging level user 6
 logging logfile messages 6
```

This is a sample output.

```
switch# bloggerd auto-collect component arp enable
Component arp auto-collect successfully enabled.
arp.yaml file copied from /bootflash/scripts/backup-staging to
/bootflash/scripts/default-autocollect
switch# dir bootflash:scripts/default-autocollect
435 Nov 10 08:43:21 2021 arp.yaml
438 Oct 25 05:55:11 2021 fex.yaml
579 Oct 25 05:55:11 2021 kern.yaml
Usage for bootflash://sup-local
11078049792 bytes used
10653151232 bytes free
21731201024 bytes total
switch# dir bootflash:scripts/backup-staging/
switch# bloggerd auto-collect component ?
  CrdCfg         Auto-collect for CRDCFG
  aclmgr         Auto-collect for ACLMgr
```

```
aclqos          Auto-collect for ACLQOS
adjmgr          Auto-collect for Adjacency Manager
arp             Auto-collect for ARP
bcm_usd         Auto-collect for BCM USD
bgp             Auto-collect for BGP
cardclient      Auto-collect for CARD CLIENT
cdp             Auto-collect for CPD
cfs             Auto-collect for CFS
clis            Auto-collect for CLIS
cts             Auto-collect for CTS
dhcp_snoop      Auto-collect for DHCP Snoop
eigrp           Auto-collect for EIGRP
eltm            Auto-collect for ELTM
ethport         Auto-collect for Eth Port Manager
feature-mgr     Auto-collect for Feature Manager
fex             Auto-collect for Fex (Satellite Manager)
hmm             Auto-collect for HMM
hsrp_engine     Auto-collect for HSRP
icam            Auto-collect for ICAM
icmpv6          Auto-collect for ICMPv6
iftmc           Auto-collect for IFTMC
im              Auto-collect for IM
ip              Auto-collect for IP
ipfib           Auto-collect for IPFIB Manager
ipqosmgr        Auto-collect for QOS Manager
isis            Auto-collect for ISIS
jer_usd         Auto-collect for JER USD
kafka           Auto-collect for KAFKA Manager
kern            Auto-collect for Kernel
l2fm            Auto-collect for L2FM
l2rib           Auto-collect for L2RIB
l3vm            Auto-collect for L3VM
lacp            Auto-collect for LACP
lldp            Auto-collect for LLDP
m2rib           Auto-collect for M2RIB
mfdm            Auto-collect for MFDM
mrib            Auto-collect for MRIB
nbm             Auto-collect for NBM Daemon
netstack        Auto-collect for Netstack
ngoam           Auto-collect for NGOAM
nve             Auto-collect for NVE
ospf            Auto-collect for Open Shortest Path First Unicast Routing Protocol (OSPF)
ospfv3        Auto-collect for Open Shortest Path First Version 3 Unicast Routing Protocol

pfma            Auto-collect for PFM
pim             Auto-collect for PIM
pktmgr          Auto-collect for Packet Manager
pltfm_config    Auto-collect for PLTFM CONFIG
port-channel    Auto-collect for Port Channel Manager
qos             Auto-collect for QOS Manager
rip             Auto-collect for RIP
sdaa            Auto-collect for SDAA
sla_responder   Auto-collect for SLA Responder
sla_sender      Auto-collect for SLA Sender
sla_twamp       Auto-collect for SLA Twamp
smm             Auto-collect for SMM
snmpmib_proc    Auto-collect for Snmpmib_proc
spm             Auto-collect for SPM
statsclient     Auto-collect for Statistics Client
sysmgr          Auto-collect for SYSMGR
tahusd          Auto-collect for TAHUSD
tctrl_usd       Auto-collect for TCTRL USD
tun_enc_mgr     Auto-collect for TEM
udld            Auto-collect for UDLD
```

```
  ufdm           Auto-collect for UFDM
  vmtracker      Auto-collect for VMTRACKER
  vntag_mgr      Auto-collect for VNTAG Mgr
  vpc            Auto-collect for VPC
  vrrp-cfg       Auto-collect for VRRP Configuration
  vrrp-eng       Auto-collect for VRRP Engine
  vrrpv3         Auto-collect for VRRPV3
Usage for bootflash://sup-local 11078049792 bytes used
10653151232 bytes free
21731201024 bytes total
switch# dir bootflash:scripts/default-autocollect^C n9k-A# dir
bootflash:scripts/default-autocollect
435 Nov 10 08:43:21 2021 arp.yaml
438 Oct 25 05:55:11 2021 fex.yaml
579 Oct 25 05:55:11 2021 kern.yaml Usage for bootflash://sup-local 11078049792 bytes used
10653151232 bytes free
21731201024 bytes total
```

This is an example to create pre-populated YAML file for the UDLD component.

```
n9k-A# bloggerd auto-collect component udld enable
Component udld auto-collect successfully enabled.
udld.yaml file copied from /bootflash/scripts/backup-staging to
/bootflash/scripts/default-autocollect
n9k-A# dir bootflash:scripts/default-autocollect
435 Nov 10 08:43:21 2021 arp.yaml
438 Oct 25 05:55:11 2021 fex.yaml
579 Oct 25 05:55:11 2021 kern.yaml
431 Nov 10 08:44:45 2021 udld.yaml
Usage for bootflash://sup-local
11078053888 bytes used
10653147136 bytes free
21731201024 bytes total
n9k-A# sh running-config all | include bloggerd
bloggerd log-dump all
bloggerd log-throttle
no bloggerd log-transfer
```

This is an example to detele pre-populated YAML file for the UDLD component.

```
n9k-A# bloggerd auto-collect component udld disable
Component udld auto-collect successfully disabled.
udld.yaml file deleted from /bootflash/scripts/default-autocollect
n9k-A# dir bootflash:scripts/default-autocollect
435 Nov 10 08:43:21 2021 arp.yaml
438 Oct 25 05:55:11 2021 fex.yaml
579 Oct 25 05:55:11 2021 kern.yaml
Usage for bootflash://sup-local
11078049792 bytes used
10653151232 bytes free
21731201024 bytes total
n9k-A#
```

Auto-Collection for protocol flap

From NX-OS Release 10.4(1)F, in case of unexpected protocol flap events, automatic data collection is enabled by default for BGP, OSPF, ISIS, and BFD protocols.

The existing auto-collect works for syslog messages of severity 0,1, and 2. Unexpected protocol flap auto collection will be triggered by routing protocol unexpected event is detected by a given protocol process. This auto-collection is controlled by system EEM policy `trigger_generic_evt_default`.

This example shows how to enable default auto collect on the BGP component.

```
switch# configure terminal
switch(config)# bloggerd auto-collect component bgp enable
component bgp auto-collect successfully enabled
bgp.yaml file copied from /bootflash/scripts/backup-staging to
/bootflash/scripts/default-autocollect
switch(config)# run bash
bash-4.4$ cd /bootflash/scripts/default-autocollect/
bash-4.4$ ls bgp.yaml
```

This example shows auto-collect history on the switch.

```
switch(config)# show system internal event-logs auto-collect history
DateTime    SnapshotID  Syslog      Status/Secs/Logsize(Bytes)
2023-Mar-21 11:04:03 1395380375    NVE-0- TEST_SYSLOGPROCESSED:23:5756541
```

Auto-Collection YAML file contents

The contents of a YAML file determine the data collected during trigger-based auto-collection. There must be only one YAML file on the switch, but it can contain auto-collection meta-data for any number of switch components and messages.

Locate the YAML file in the `/bootflash/scripts` directory on the switch.

Invoke the YAML file for trigger-based collection by using the following example. The example shows the minimum required configuration for trigger-based collection to work with a user-defined YAML file.

```
switch# show running-config eem
!Command: show running-config eem
!Running configuration last done at: Mon Sep 30 19:34:54 2019
!Time: Mon Sep 30 22:24:55 2019
version 9.3(3) Bios:version 07.59
event manager applet test_1 override __syslog_trigger_default
  action 1.0 collect test.yaml $_syslog_msg
```

In the preceding example, `test_1` is the name of the applet and `test.yaml` is the name of the user-configured YAML file present in the `/bootflash/scripts` directory.

Example YAML file

This is an example of a basic YAML file supporting the trigger-based event log auto-collection feature. The definitions for the keys/values in the file are in the table.

Note Make sure that the YAML file has proper indentation. As a best practice, run it through any `online YAML validator` before using it on a switch.

```
bash-4.3$ cat /bootflash/scripts/test.yaml
version: 1
components:
   securityd:
        default:
            tech-sup: port
            commands: show module
   platform:
        default:
            tech-sup: port
            commands: show module
```

Key: Value	Description
version: 1	Set to 1. Any other number creates an incompatibility for the auto collect script.
components:	Keyword specifying that what follows are switch components.
securityd:	Name of the syslog component (`securityd` is a facility name in syslog).
default:	Identifies all messages belonging to the component.
tech-sup: port	Collect tech support of the port module for the `securityd` syslog component.
commands: show module	Collect show module command output for the `securityd` syslog component.
platform:	Name of the syslog component (`platform` is a facility name in syslog).
tech-sup: port	Collect tech support of the port module for the `platform` syslog component.
commands: show module	Collect show module command output for the `platform` syslog component.

Use this example to associate auto-collect metadata only for a specific log, for example, SECURITYD-2-FEATURE_ENABLE_DISABLE.

```
securityd:
               feature_enable_disable:
                       tech-sup: security
                       commands: show module
```

Key: Value	Description
securityd:	Name of the syslog component (`securityd` is a facility name in syslog).
feature_enable_disable:	Message ID of the syslog message.
tech-sup: security	Collect tech support of the security module for the `securityd` syslog component.
commands: show module	Collect show module command output for the security syslog component.

This is an example syslog output for the above YAML entry.

```
2019 Dec 4 12:41:01 n9k-c93108tc-fx %SECURITYD-2-FEATURE_ENABLE_DISABLE: User
has enabled the feature bash-shell
```

Use this example to specify multiple values.

```
version: 1
components:
   securityd:
        default:
            commands: show module;show version;show module
            tech-sup: port;lldp
```

Note Use semicolons to separate multiple show commands and tech support key values (see the preceding example).

Beginning with Release 10.1(1), `test.yaml` can be replaced with a folder inside which more than one YAML files can be present. All the YAML files in the folder must follow the `ComponentName.yaml` naming convention.

In this example, `test.yaml` is replaced with `test_folder`.

```
test.yaml:
event manager applet logging2 override __syslog_trigger_default
   action 1.0 collect test.yaml rate-limt 30 $_syslog_msg

test_folder:
event manager applet logging2 override __syslog_trigger_default
   action 1.0 collect test_folder rate-limt 30 $_syslog_msg
```

This example shows the path and component(s) for `test_folder`.

```
ls /bootflash/scripts/test_folder
bgp.yaml ppm.yaml
```

Auto-collection bundle limits per component

This topic describes the default limit of auto-collection bundles per component event and the behavior when the limit is reached.

For auto-collection, the system limits the number of bundles saved per component event. This is set to one (1) by default from NX-OS Release 10.2(2)F. Earlier, this limit was three (3) by default. If more events occur for a component than the default set for the component, then the events are dropped with the status message `EVENTLOGLIMITREACHED`. Auto-collection resumes for the component after the event log has rolled over.

Example:

```
switch# show system internal event-logs auto-collect history
DateTime              Snapshot ID  Syslog                   Status/Secs/Logsize(Bytes)
2020-Jun-27 07:20:03  1140276903   ACLMGR-0-TEST_SYSLOG     EVENTLOGLIMITREACHED
2020-Jun-27 07:15:14  1026359228   ACLMGR-0-TEST_SYSLOG     RATELIMITED
2020-Jun-27 07:15:09  384952880    ACLMGR-0-TEST_SYSLOG     RATELIMITED
2020-Jun-27 07:13:55  1679333688   ACLMGR-0-TEST_SYSLOG     PROCESSED:2:9332278
2020-Jun-27 07:13:52  1679333688   ACLMGR-0-TEST_SYSLOG     PROCESSING
2020-Jun-27 07:12:55  502545693    ACLMGR-0-TEST_SYSLOG     RATELIMITED
2020-Jun-27 07:12:25  1718497217   ACLMGR-0-TEST_SYSLOG     RATELIMITED
2020-Jun-27 07:08:25  1432687513   ACLMGR-0-TEST_SYSLOG     PROCESSED:2:10453823
2020-Jun-27 07:08:22  1432687513   ACLMGR-0-TEST_SYSLOG     PROCESSING
2020-Jun-27 07:06:16  90042807     ACLMGR-0-TEST_SYSLOG     RATELIMITED
2020-Jun-27 07:03:26  1737578642   ACLMGR-0-TEST_SYSLOG     RATELIMITED
2020-Jun-27 07:02:56  40101277     ACLMGR-0-TEST_SYSLOG     PROCESSED:3:10542045
2020-Jun-27 07:02:52  40101277     ACLMGR-0-TEST_SYSLOG     PROCESSING
```

Auto-Collection log files

Auto-Collection log files are generated and managed based on YAML configuration, with options to control purge frequency and storage locations.

About Auto-Collection log files

The configuration in a YAML file determines the contents of an auto-collected log file. You can't configure the amount of memory used for collected log files. You can configure the frequency of when the stored files get purged.

Auto-collected log files get saved in the following directory:

```
switch# dir bootflash:eem_snapshots
   44205843    Sep 25 11:08:04 2019
1480625546_SECURITYD_2_FEATURE_ENABLE_DISABLE_eem_snapshot.tar.gz
  Usage for bootflash://sup-local
 6940545024 bytes used
44829761536 bytes free
51770306560 bytes total
```

Accessing the log files

Locate the logs by using the command keyword `debug`.

```
switch# dir debug:///
...
       26    Oct 22 10:46:31 2019  log-dump
       24    Oct 22 10:46:31 2019  log-snapshot-auto
       26    Oct 22 10:46:31 2019  log-snapshot-user
```

This table describes the log locations and the log types stored.

Table 25: Log Locations and Types

Location	Description
log-dump	This folder stores Event logs on log rollover.
log-snapshot-auto	This folder contains the auto-collected logs for syslog events 0, 1, 2.
log-snapshot-user	This folder stores the collected logs when you run the `bloggerd log-snapshot <>` command.

Use this example to view the log files generated on log rollover.

```
switch# dir debug:log-dump/
debug:log-dump/20191022104656_evtlog_archive.tar
debug:log-dump/20191022111241_evtlog_archive.tar
debug:log-dump/20191022111841_evtlog_archive.tar
debug:log-dump/20191022112431_evtlog_archive.tar
debug:log-dump/20191022113042_evtlog_archive.tar
debug:log-dump/20191022113603_evtlog_archive.tar
```

Parse the log tar files

Use this example to parse the logs in the tar files.

```
switch# show system internal event-logs parse debug:log-dump/20191022104656_evtlog_archive.tar
--------LOGS:/tmp/BLOGGERD0.991453012199/tmp/1-191022104658-191022110741-device_test-M27-V1-I1:0-P884.gz--------
2019 Oct 22 11:07:41.597864 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):Data Space
Limits(bytes): Soft: -1  Ha rd: -1
2019 Oct 22 11:07:41.597857 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):Stack Space
Limits(bytes): Soft: 500000  Hard: 500000
2019 Oct 22 11:07:41.597850 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):AS: 1005952076
-1
2019 Oct 22 11:07:41.597406 E_DEBUG Oct 22 11:07:41 2019(device_test_process_events):Sdwrap
 msg unknown
2019 Oct 22 11:07:41.597398 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):Going back to
select
2019 Oct 22 11:07:41.597395 E_DEBUG Oct 22 11:07:41 2019(nvram_test):TestNvram examine 27
blocks
2019 Oct 22 11:07:41.597371 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):Parent: Thread
created test index:4 thread_id:-707265728
```

```
2019 Oct 22 11:07:41.597333 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):Node inserted
2019 Oct 22 11:07:41.597328 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):The test index
in diag is 4
2019 Oct 22 11:07:41.597322 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):result severity
level
2019 Oct 22 11:07:41.597316 E_DEBUG Oct 22 11:07:41 2019(diag_test_start):callhome alert
level
```

This table describes the additional keywords available for parsing the specific tar file.

Table 26: Parse Keywords

Keyword	Description
component	Decode logs belonging to the component identified by process name.
from-datetime	Decode logs from a specific date and time in yy[mm[dd[HH[MM[SS]]]]] format.
instance	List of SDWRAP buffer instances to be decoded (comma separated).
module	Decode logs from modules such as SUP and LC (using module IDs).
to-datetime	Decode logs up to a specific date and time in yy[mm[dd[HH[MM[SS]]]]] format.

Copy logs to a different location

Use this example to copy logs to a different location such as a remote server.

```
switch# copy debug:log-dump/20191022104656_evtlog_archive.tar
scp://<ip-adress>/nobackup/<user> vrf management  use-kstack
Enter username: user@<ip-address>'s password:
20191022104656_evtlog_archive.tar                                          100%  130KB
130.0KB/s   00:00
Copy complete, now saving to disk (please wait)...
Copy complete.
```

Purge Auto-Collection Log Files

There are two types of generated trigger-based auto-collection logs: EventHistory and EventBundle.

Purge logic for EventHistory logs

For event history, purging occurs in the `/var/sysmgr/srv_logs/xport` folder. 250MB of partitioned RAM is mounted at the `/var/sysmgr/srv_logs` directory.

If the `/var/sysmgr/srv_logs` memory usage is under 65% of the 250MB allocated, no files get purged. When the memory utilization reaches the 65% limit level, the oldest files get purged until there's enough memory available to continue saving new logs.

Purge logic for EventBundle logs

For event bundles, the purge logic occurs in the `/bootflash/eem_snapshots` folder. For storing the auto-collected snapshots, the Embedded Event Manager (EEM) auto-collect script allocates 5% of the bootflash storage. The logs get purged once the 5% bootflash capacity is used.

When a new auto-collected log is available but there is no space to save it in bootflash (already at 5% capacity), the system checks:

1. If there are existing auto-collected files that are more than 12 hours old, the system deletes the files and the new logs get copied.
2. If the existing auto collected files are less than 12 hours old, the system discards the newly collected logs without saving them.

You can modify the 12-hour default purge time by using the following commands. The time specified in the command is in minutes.

```
switch(config)# event manager applet test override __syslog_trigger_default
switch(config-applet)# action 1.0 collect test.yaml purge-time 300 $_syslog_msg
```

event manager command: *test* is an example name for the policy. **__syslog_trigger_default** is the name of the system policy that you want to override. This name must begin with a double underscore (__).

action command: **1.0** is an example number for the order in which the action is executed. **collect** indicates that data is collected using the YAML file. *test.yaml* is an example name of the YAML file. **$_syslog_msg** is the name of the component.



Note At any given time, there can be only one trigger-based auto-collection event in progress. If another new log event is attempting to be stored when auto-collection is already occurring, the new log event is discarded.

By default, there's only one trigger-based bundle collected every five minutes (300 sec). This rate limiting is also configurable by the following commands. The time specified in the command is in seconds.

```
switch(config)# event manager applet test override __syslog_trigger_default
switch(config-applet)# action 1.0 collect test.yaml rate-limit 600 $_syslog_msg
```

event manager command: *test* is an example name for the policy. **__syslog_trigger_default** is an example name of the system policy to override. This name must begin with a double underscore (__).

action command: **1.0** is an example number for the order in which the action is executed. **collect** indicates that data is collected using the YAML file. *test.yaml* is an example name of the YAML file. **$_syslog_msg** is the name of the component.

Beginning with Release 10.1(1), the rate of collection can also be regulated using a maximum number of triggers option, ensuring that only those many number of triggers are honored. After the **max-triggers** value is reached, no more bundles will be collected on the syslog occurrence.

```
event manager applet test_1 override __syslog_trigger_default
  action 1.0 collect test.yaml rate-limt 30 max-triggers 5 $_syslog_msg
```



Note If you delete auto collected bundles manually from `debug:log-snapshot-auto/`, then it will restart the collection based on the configured number of **max-triggers** when the next event occurs.

Auto-Collection statistics and history

This example shows trigger-based collection statistics.

```
switch# show system internal event-logs auto-collect statistics
--------------------EEM Auto Collection Statistics--------------------
Syslog Parse Successful :88 Syslog Parse Failure :0
Syslog Ratelimited :0 Rate Limit Check Failed :0
Syslog Dropped(Last Action In Prog) :53 Storage Limit Reached :0
User Yaml Action File Unavailable :0 User Yaml Parse Successful :35
User Yaml Parse Error :0 Sys Yaml Action File Unavailable :11
Sys Yaml Parse Successful :3 Sys Yaml Parse Error :0
Yaml Action Not Defined :0 Syslog Processing Initiated :24
Log Collection Failed :0 Tar Creation Error :0
Signal Interrupt :0 Script Exception :0
Syslog Processed Successfully :24 Logfiles Purged :0
```

This example shows trigger-based collection history (the processed syslogs, process time, size of the data collected) obtained using a command line interface command.

```
switch# show system internal event-logs auto-collect history
DateTime Snapshot ID Syslog Status/Secs/Logsize(Bytes)
2019-Dec-04 05:30:32 1310232084 VPC-0-TEST_SYSLOG PROCESSED:9:22312929
2019-Dec-04 05:30:22 1310232084 VPC-0-TEST_SYSLOG PROCESSING
2019-Dec-04 04:30:13 1618762270 ACLMGR-0-TEST_SYSLOG PROCESSED:173:33194665
2019-Dec-04 04:28:47 897805674 SYSLOG-1-SYSTEM_MSG DROPPED-LASTACTIONINPROG
2019-Dec-04 04:28:47 947981421 SYSLOG-1-SYSTEM_MSG DROPPED-LASTACTIONINPROG
2019-Dec-04 04:27:19 1618762270 ACLMGR-0-TEST_SYSLOG PROCESSING
2019-Dec-04 02:17:16 1957148102 CARDCLIENT-2-FPGA_BOOT_GOLDEN NOYAMLFILEFOUND
```

Examples of supported log collection

A sample collection of supported logs for a few components are:

- Component: IPQoSMgr

  Supported logs:

  ```
  QOSMGR_MTS_FAILURE
  QOSMGR_NETWORK_QOS_POLICY_CHANGE
  QOSMGR_LLFC_APPLY_FAILURE
  QOSMGR_FCOE_POLICY_NOT_REMOVED
  ```

- Component: ACLQOS

  Supported logs:

  ```
  ACLQOS_UNEXPECTED_MCAST_FRAMES
  ACLQOS_UNEXPECTED_PFC_FRAMES
  PPF_SUBSCRIPTION_FAILED
  ACLQOS_QOS_NO_DROP_CLASSIFICATION_UNSUPPORTED
  ACLQOS_QUEUE_LIMIT_IGNORED_ON_FEX
  ACLQOS_BUFFER_DRAIN_FAILURE
  ACLQOS_BURST_DETECT_FPGA_INCOMPATIBLE
  ACLQOS_BURST_DETECT_OVER_THRESHOLD
  ACLQOS_FAILED
  PPF_FAILED
  ```

Verify trigger-based log collection

Procedure

Verify that the trigger-based log collection feature is enabled by entering the **show event manager system-policy | i trigger** command, and here is an example.

Example:

```
switch# show event manager system-policy | i trigger n 2
           Name : __syslog_trigger_default
    Description : Default policy for trigger based logging
    Overridable : Yes
     Event type : 0x2101
```

Check trigger-based log file generation

You can check to see if the trigger-based auto-collection feature has generated any event log files.

You can check to see if the trigger-based auto-collection feature has generated any event log files. Enter one of the commands in these examples:

```
switch# dir bootflash:eem_snapshots
9162547 Nov 12 22:33:15 2019 1006309316_SECURITYD_2_FEATURE_ENABLE_DISABLE_eem_snapshot.tar.gz

Usage for bootflash://sup-local
8911929344 bytes used
3555950592 bytes free
12467879936 bytes total
```

```
switch# dir debug:log-snapshot-auto/
63435992 Dec 03 06:28:52 2019
20191203062841_1394408030_PLATFORM_2_MOD_PWRDN_eem_snapshot.tar.gz

Usage for debug://sup-local
544768 bytes used
4698112 bytes free
5242880 bytes total
```

Local log file storage

Local log file storage capabilities are:

- Amount of local data storage time depends on the scale, and type, of deployment. For both modular and nonmodular switches, the storage time is from 15 minutes to several hours of data. To be able to collect relevant logs that span a longer period:
  - Only enable event log retention for the specific services/features you need. See Enable Extended Log File Retention for a single service, on page 330.
  - Export the internal event logs off the switch. See External log file storage, on page 349.
- Compressed logs are stored in RAM.
- 250MB memory is reserved for log file storage.
- Log files are optimized in tar format (one file for every five minutes or 10MB, whichever occurs first).
- Allow snap-shot collection.

Generate a local copy of recent log files

Extended Log File Retention is enabled by default for all services running on a switch. Log files are stored locally on flash memory.

Procedure

Create a snapshot bundle file of the last ten event logs stored on the switch using the **bloggerd log-snapshot** [*file-name*] [**bootflash:** *file-path* | **logflash:** *file-path* | **usb1:**] [**size** *file-size*] [**time** *minutes*] command.

Example:

```
switch# bloggerd log-snapshot snapshot1
```

Default storage for this operation is **logflash**.

file-name: The filename of the generated snapshot log file bundle. Use a maximum of 64 characters for *file-name*.



Note This variable is optional. If it is not configured, the system applies a timestamp and `_snapshot_bundle.tar` as the filename. Example:

```
20200605161704_snapshot_bundle.tar
```

bootflash: *file-path*: The file path where the snapshot log file bundle is being stored on the bootflash. Choose one of the following initial paths:

- bootflash:///
- bootflash://module-1/
- bootflash://sup-1/
- bootflash://sup-active/
- bootflash://sup-local/

logflash: *file-path*: The file path where the snapshot log file bundle is being stored on the logflash. Choose one of the following initial paths:

- logflash:///
- logflash://module-1/
- logflash://sup-1/
- logflash://sup-active/
- logflash://sup-local/

usb1:: The file path where the snapshot log file bundle is being stored on the USB device.

size *file-size*: The snapshot log file bundle based on size in megabytes (MB). Range is from 5MB through 250MB.

time *minutes*: The snapshot log file bundle based on the last x amount of time (minutes). Range is from 1 minute through 30 minutes.

```
switch# bloggerd log-snapshot snapshot1
Snapshot generated at logflash:evt_log_snapshot/snapshot1_snapshot_bundle.tar Please cleanup
```

```
 once done.
switch#
switch# dir logflash:evt_log_snapshot
159098880 Dec 05 06:40:24 2019 snapshot1_snapshot_bundle.tar
159354880 Dec 05 06:40:40 2019 snapshot2_snapshot_bundle.tar

Usage for logflash://sup-local
759865344 bytes used
5697142784 bytes free
6457008128 bytes total
```

Display the same files using the command in this example:

```
switch# dir debug:log-snapshot-user/
159098880 Dec 05 06:40:24 2019 snapshot1_snapshot_bundle.tar
159354880 Dec 05 06:40:40 2019 snapshot2_snapshot_bundle.tar

Usage for debug://sup-local
929792 bytes used
4313088 bytes free
5242880 bytes total
```



Note Note the filename at the end of the example. Each individual log file is also identified by the date and time it was generated.

The LC core file includes the `log-snapshot` bundle. The `log-snapshot` bundle filename is `tac_snapshot_bundle.tar.gz`. Here is an example:

```
bash-4.2$ tar -tvf 1610003655_0x102_aclqos_log.17194.tar.gz
drwxrwxrwx root/root 0 2021-01-07 12:44 pss/
-rw-rw-rw- root/root 107 2021-01-07 12:44 pss/dev_shm_aclqos_runtime_info_lc.gz
-rw-rw-rw- root/root 107 2021-01-07 12:44 pss/dev_shm_aclqos_runtime_cfg_lc.gz
-rw-rw-rw- root/root 107 2021-01-07 12:44 pss/dev_shm_aclqos_debug.gz
-rw-rw-rw- root/root 129583 2021-01-07 12:44 pss/clqosdb_ver1_0_user.gz
-rw-rw-rw- root/root 20291 2021-01-07 12:44 pss/clqosdb_ver1_0_node.gz
-rw-rw-rw- root/root 444 2021-01-07 12:44 pss/clqosdb_ver1_0_ctrl.gz
drwxrwxrwx root/root 0 2021-01-07 12:44 proc/
-rw-rw-rw- root/root 15159 2021-01-07 12:44 0x102_aclqos_compress.17194.log.25162
-rw-rw-rw- root/root 9172392 2021-01-07 12:43 0x102_aclqos_core.17194.gz
-rw-rw-rw- root/root 43878 2021-01-07 12:44 0x102_aclqos_df_dmesg.17194.log.gz
-rw-rw-rw- root/root 93 2021-01-07 12:44 0x102_aclqos_log.17194
-rw-rw-rw- root/root 158 2021-01-07 12:44 0x102_aclqos_mcore.17194.log.gz
drwxrwxrwx root/root 0 2021-01-07 12:44 usd17194/
-rw-rw-rw- root/root 11374171 2021-01-07 12:44 tac_snapshot_bundle.tar.gz
```

External log file storage

External log file storage is a solution that

- securely stores switch logs on an external server through HTTPS,
- supports about 10 switches per server, and
- requires manual certificate management.

Characteristics and operational details of external log file storage

External log file storage allows you to enable secure, off-switch storage of logs using an external server.

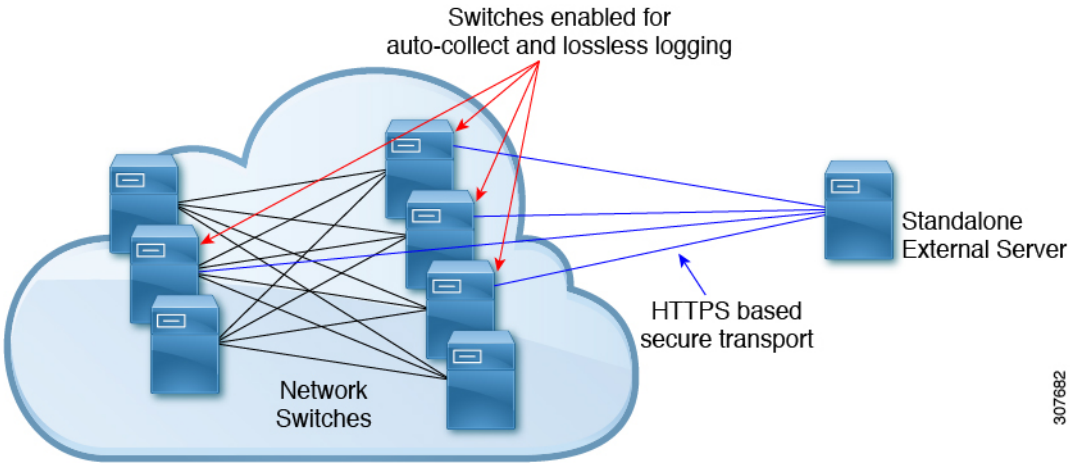
- Logs are stored securely outside the switch.
- Supports HTTPS-based transport for log transfer.
- Storage requirements differ by switch type: 300MB for nonmodular switches, 12GB per day per modular switch.
- External server typically stores logs for 10 switches, but there is no firm limit.

An external server solution provides the capability to store logs off-switch in a secure manner.

Note To create the external storage capability, contact Technical Assistance Center(TAC) to help deploy the external server solution.

The external log file storage capabilities are:

- Enabled on-demand
- HTTPS-based transport
- Storage requirements:
  - Nonmodular switches: 300MB
  - Modular switches: 12GB (per day, per switch)
- An external server generally stores logs for 10 switches. However, there's no firm limit to the number of switches supported by an external server.



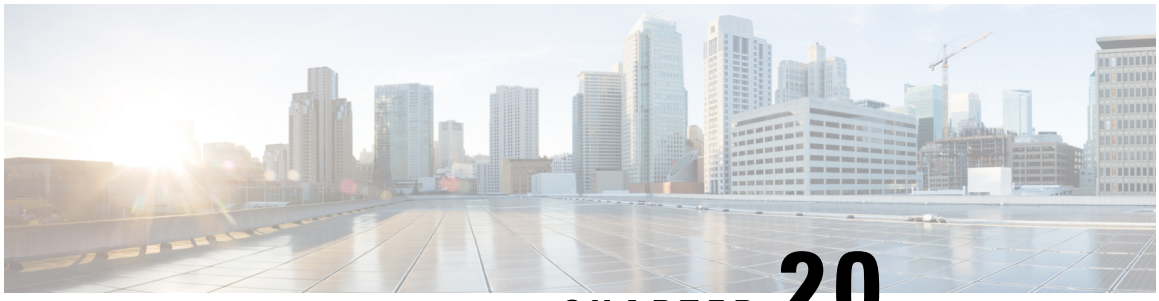
The external server solution has the following characteristics:

- Controller-less environment
- Manual management of security certificates

- Three supported use-cases:
  - Continuous collection of logs from selected switches
  - TAC-assisted effort to deploy and upload logs to servers.
  - Limited on-premise processing
- Data are exported using telemetry



Note Contact TAC for information regarding the setup and collection of log files in an external server.



CHAPTER 20

Configuring MAC Move Policy

This chapter describes how to configure the MAC move policy on Cisco NX-OS devices.

This chapter includes the following sections:

- MAC move policy, on page 353
- Guidelines and Limitations for MAC Move Policy, on page 353
- Configuring MAC Move Policy, on page 354
- Verifying the MAC Move Policy Configuration, on page 355

MAC move policy

The MAC move policy feature is used to monitor the movement of MAC addresses across the fabric.

This feature provides the option to modify the MAC move parameters in order to enhance network stability and prevent the accumulation of Messaging and Transaction Service (MTS) in the Layer 2 Forwarder Manipulator (L2FM).

By default, the MAC move policy is disabled. For more information on configuring the MAC move policy and enabling the MTS buildup check, see Configuring MAC Move Policy, on page 354 section.

The advantages of MAC move policy are:

- Enables tracking of MAC address movement at the MAC level to avoid loops.
- Checks periodically for MTS buildup to prevent crashes in the L2FM.
- Disables VLAN learning if the MAC move count surpasses the specified threshold.

Guidelines and Limitations for MAC Move Policy

MAC move Policy has the following configuration guidelines and limitations:

- Beginning with Cisco NX-OS Release 10.3(1)F, the MAC move policy is supported on Cisco Nexus 9300-X Cloud Scale switches.
- When the MAC move policy is enabled using the **mac-move policy** command, the default parameters (6 MAC moves in 30 seconds with a hold interval of 120 seconds) are set. You can modify the default set parameters. For more information, see Configuring MAC Move Policy, on page 354.

- MAC move policy and the L2RIB detection cannot coexist with their default values. Both these mechanisms have a different approach to handle duplicate detection.
- If the MAC move policy is not required in a VXLAN environment, do not enable it. If it's required, make sure to modify either the L2RIB policy or the MAC move policy default value.
- L2RIB detection can be changed using the **l2rib dup-host-mac-detection <mac moves threshold> <detect-interval>** command.
- The following behavior is observed when both MAC move policy and L2RIB detection are configured:
  - When L2RIB detection is less than L2 policy, only L2RIB detection will be triggered. However, the L2 policy is not triggered.
  - When L2RIB detection is equal to L2 policy, either L2 policy or L2RIB detection will be triggered. However, there is no guarantee of the order in which these policies will be triggered.
  - When L2RIB detection is greater than L2 policy, only L2 policy will be triggered. However, the L2RIB detection is not triggered.

Configuring MAC Move Policy

This procedure enables/disables the MAC move policy on the switch:

SUMMARY STEPS

1. **configure terminal**
2. **[no] mac-move policy {[move-threshold**<*thresh*>]} {[**detect-intvl**<*d_intvl*>]} {[**hold-intvl** <*h_intvl*>]}
3. **[no] mts-buildup check** {[**mts-percent**<*percent*>]} {[**detect-intvl**<*d_intvl*>]}

DETAILED STEPS

Procedure

	Command or Action	Purpose

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

Step 2	**[no] mac-move policy {[move-threshold**<*thresh*>]} {[**detect-intvl**<*d_intvl*>]} {[**hold-intvl** <*h_intvl*>]} **Example:** `switch(config)# mac-move policy move-threshold 6 detect-intvl 30 hold-intvl 120`	Enables the MAC move policy. The option **no** disables the MAC move policy. • *thresh*: Maximum allowed moves. The range is 2–4096 and the default value is value 6. • *d_intvl*: Interval to track moves. The range is 30–3600 and the default value is 30.

	Command or Action	Purpose
		h_intvl: Interval to disable learning. The range is 120–360 and the default value is 120.
Step 3	**[no] mts-buildup check** {[**mts-percent**<*percent*>]} {[**detect-intvl**<*d_intvl*>]} **Example:** `switch(config)# mts-buildup check mts-percent 40 detect-intvl 60`	Enables the mts-buildup check. The option **no** disables the mts-buildup check. • *percent*: MTS buildup percent value. The range is 10–50 and the default value is 40. • *d_intvl*: Interval to check MTS buildup. The range is 60-600 and the default value is 60.

Verifying the MAC Move Policy Configuration

To display mac-move policy configuration information, perform one of the following tasks:

Command	Purpose
show mac-move policy	Displays information about the mac-move policy.
show mts-buildup check	Displays information about the mts-buildup check.

The following example shows sample output for the **show mac-move policy** command:

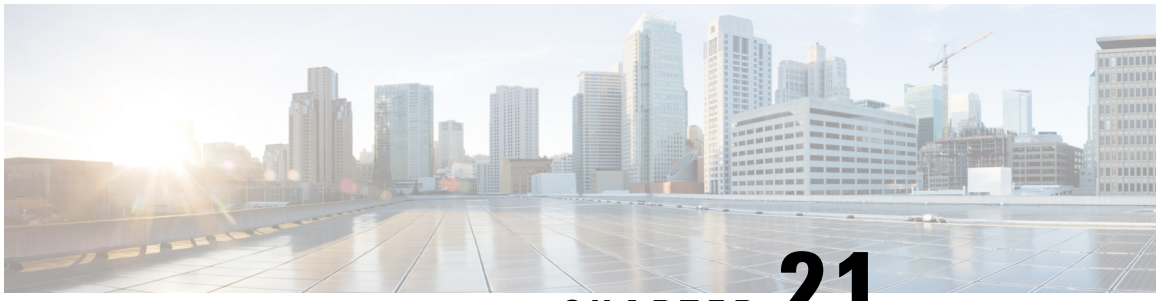
```
switch(config)# show mac-move policy
MAC move policy enabled = TRUE
MAC move policy threshold = 6
MAC move policy detect interval = 30
MAC move policy hold interval =120
switch(config)#
```

The following example shows sample output for the **show mts-buildup check** command when check is enabled:

```
switch(config)# show mts-buildup check
MTS buildup check enabled =TRUE
MTS check percent =40
MTS check interval =60
switch(config)#
```

The following example shows sample output for the **show mts-buildup check** command when check is disabled:

```
switch(config)# show mts-buildup check
MTS buildup check enabled =FALSE
ppwrks5(config)# show mac-move policy
MAC move policy enabled = FALSE
switch(config)#
```



CHAPTER 21

Terminal Lock for VSH Sessions

- Overview, on page 357

Overview

Currently with NX-OS, there are many users logged in to the switch and make configuration changes in their sessions with CLI. The goal is to restrict this scenario and allow only one user to configure the switch. This is achieved by terminal lock CLIs that lock the terminal to allow only one user to access the configure terminal commands. As a result, the effect of a “configuration lock” is achieved that prevents other users from changing the NX-OS running configuration.

Terminal lock feature provides a locking mechanism to enable users to have an exclusive configuration access to modify NX-OS running configuration.

The following is the sequence of operations:

1. `terminal lock` – This CLI provides the configuration lock to the user.
2. `terminal unlock` – This CLI releases the terminal lock taken by any session.
3. `show terminal lock` – This CLI shows the current terminal lock status and details.

Terminal Lock

The following are the guidelines for the terminal lock usage:

- `terminal lock` allows config commands to be executed only in that current session where the lock is held.
- `terminal lock` blocks only config commands in the other sessions, that means SHOW or EXEC CLIs are still allowed.
- Default timeout for terminal lock is 1800 seconds (30 minutes).
- Once the lock timer expires, terminal lock is released automatically.
- `terminal lock` CLI can be executed by any user with network-admin privilege.
- `terminal lock` is rejected if "configure dual-stage" session is in progress.

The following is the example CLI for the terminal lock:

```
switch# terminal lock?
lock Locks the CLI Config mode
switch# terminal lock ?
<CR>
<60-43200> Enter terminal lock timeout in seconds
*Default value is 1800
“terminal lock” locks the parser configuration mode and prints a syslog message as shown
in below example.
switch# terminal lock
switch# 2021 Jun 19 17:53:37 switch %VSHD-5-VSHD_CLI_TERM_LOCK: terminal lock is taken by
admin on console0
```



Note If a user tries to enter a configured terminal in another session, the following error message is displayed: “Configuration locked. terminal lock is taken by other VSH session.”

Beginning with Cisco NX-OS Release 10.2(2)F, a new CLI option, **"terminal lock mdp"** is introduced to lock Model Driven Programmability interfaces like RESTCONF, NETCONF, gRPC, gNMI, and so on.

The CLI **"terminal lock mdp"** makes the terminal lock applicable to all configuration sessions including DME sessions.

The following is the sample output for the **"terminal lock mdp"** CLI:

```
switch# terminal lock?
  lock  Locks the CLI Config mode
switch# terminal lock ?
  <CR>
  <mdp>  Locks Model Driven Programmability sessions
  <60-43200>  Enter terminal lock timeout in seconds
              *Default value is 1800

switch# terminal lock mdp
2021 Oct 26 06:33:19 switch %VSHD-5-VSHD_CLI_TERM_LOCK: terminal lock is taken by admin on
 console0
switch#
switch# show terminal lock
PID: 10018
User: admin
Session: console0
State: LOCKED
MDP lock: True
Lock acquired time: Mon Mar  8 09:24:03 2021
Lock Expiration timer (in Sec): 1800
switch#
```

Terminal Unlock

The following is the example CLI for the terminal unlock:

```
switch# terminal unlock?
unlock Force unlocking of the CLI config mode
switch# terminal unlock ?
<CR>
switch# terminal unlock
switch# 2021 Jun 19 17:53:21 switch %VSHD-5-VSHD_CLI_TERM_LOCK: terminal lock is released
by admin on console0
```



Note "terminal lock" can be taken by only one admin user, but lock can be released by any admin user using "terminal unlock."

Show Terminal Lock

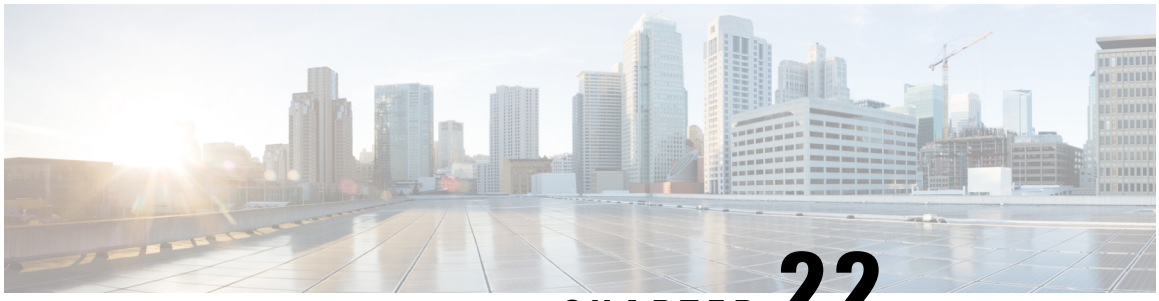
This command displays the status and details of any current configuration locks, including the owner, user, session, lock state, and lock timer.

The following is the example CLI for the Show Terminal Lock when the lock is active:

```
switch# terminal lock
switch#
switch# show terminal lock
PID: 10018
User: admin
Session: console0
State: LOCKED
Lock acquired time: Mon Mar 8 09:24:03 2021
```

The following is the example CLI for the Show Terminal Lock when the lock is free:

```
switch# terminal unlock
switch#
switch#
switch# show terminal lock
PID: -1
User: unknown
Session: NA
State: FREE
Lock acquired time:
Lock Expiration timer (in Sec): 0
switch#
```



CHAPTER 22

Onboard Failure Logging

Cisco NX-OS provides the ability to log failure data to persistent storage, which you can retrieve and display for analysis at a later time. This onboard failure logging (OBFL) feature stores failure and environmental information in nonvolatile memory on the module. The information will help analyze failed modules.

- Overview, on page 361
- Guidelines and Limitations, on page 362
- Configuration, on page 362
- Related Documents, on page 365

Overview

OBFL (On-Board Failure Logging) is a feature that stores critical operational and diagnostic data for hardware modules.

Types of Data Stored by OBFL

OBFL stores the following types of data:

- Time of initial power-on
- Slot number of the module in the chassis
- Initial temperature of the module
- Firmware, BIOS, FPGA, and ASIC versions
- Serial number of the module
- Stack trace for crashes
- CPU hog information
- Memory leak information
- Software error messages
- Hardware exception logs
- Environmental history
- OBFL-specific history information

- ASIC interrupt and error statistics history
- ASIC register dumps

Default Settings

The following table lists the default settings for OBFL parameters.

Parameters	Default
OBFL	All features enabled

Guidelines and Limitations

Prerequisites

You must have network-admin user privileges.

General Guidelines

OBFL has the following guidelines and limitations:

- OBFL is enabled by default.
- OBFL flash supports a limited number of writes and erases. The more logging that you enable, the faster you use up this number of writes and erases.

Note Be aware that the Cisco NX-OS commands for this feature may differ from those commands that are used in Cisco IOS.

Configuration

You can configure the OBFL features on Cisco NX-OS devices.

Before you begin

Make sure that you are in global configuration mode.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Enable all OBFL features using the command **hw-module logging onboard**

Example:

```
switch(config)# hw-module logging onboard
 Module: 7 Enabling ... was successful.
 Module: 10 Enabling ... was successful.
 Module: 12 Enabling ... was successful.
```

Step 3 Enable the OBFL counter statistics using the command **hw-module logging onboard counter-stats**

Example:

```
switch(config)# hw-module logging onboard counter-stats
 Module: 7 Enabling counter-stats ... was successful.
 Module: 10 Enabling counter-stats ... was successful.
 Module: 12 Enabling counter-stats ... was successful.
```

Step 4 Enable the OBFL CPU hog events using the command **hw-module logging onboard cpuhog**

Example:

```
switch(config)# hw-module logging onboard cpuhog
 Module: 7 Enabling cpu-hog ... was successful.
 Module: 10 Enabling cpu-hog ... was successful.
 Module: 12 Enabling cpu-hog ... was successful.
```

Step 5 Enable the OBFL environmental history using the command **hw-module logging onboard environmental-history**

Example:

```
switch(config)# hw-module logging onboard environmental-history
 Module: 7 Enabling environmental-history ... was successful.
 Module: 10 Enabling environmental-history ... was successful.
 Module: 12 Enabling environmental-history ... was successful.
```

Step 6 Enable the OBFL error statistics using the command **hw-module logging onboard error-stats**

Example:

```
switch(config)# hw-module logging onboard error-stats
 Module: 7 Enabling error-stats ... was successful.
 Module: 10 Enabling error-stats ... was successful.
 Module: 12 Enabling error-stats ... was successful.
```

Step 7 Enable the OBFL interrupt statistics using the command **hw-module logging onboard interrupt-stats**

Example:

```
switch(config)# hw-module logging onboard interrupt-stats
      Module: 7 Enabling interrupt-stats ... was successful.
      Module: 10 Enabling interrupt-stats ... was successful.
      Module: 12 Enabling interrupt-stats ... was successful.
```

Step 8 Enable the OBFL information for a module using the command **hw-module logging onboard module** *slot* and enable the boot uptime, device version, and OBFL history using the command **hw-module logging onboard obfl-logs**

Example:

```
switch(config)# hw-module logging onboard module 7
 Module: 7 Enabling ... was successful.
```

Example:

```
switch(config)# hw-module logging onboard obfl-logs
      Module: 7 Enabling obfl-log ... was successful.
      Module: 10 Enabling obfl-log ... was successful.
      Module: 12 Enabling obfl-log ... was successful.
```

Step 9 (Optional) Display information about OBFL using the command **show logging onboard** and copy the running configuration to the startup configuration using the command **copy running-config startup-config** .

Example:

```
switch(config)# show logging onboard
```

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Note To display OBFL information stored in flash on a module, see Verifying the OBFL Configuration .

This example shows how to enable OBFL on module 2 for environmental information:

```
switch# configure terminal
switch(config)# hw-module logging onboard module 2 environmental-history
```

What to do next

To display OBFL information stored in flash on a module, perform one of the following tasks:

Command	Purpose
show logging onboard boot-uptime	Displays the boot and uptime information.
show logging onboard counter-stats	Displays statistics on all ASIC counters.
show logging onboard credit-loss	Displays OBFL credit loss logs.
show logging onboard device-version	Displays device version information.
show logging onboard endtime	Displays OBFL logs to a specified end time.
show logging onboard environmental-history	Displays environmental history.
show logging onboard error-stats	Displays error statistics.
show logging onboard exception-log	Displays exception log information.
show logging onboard interrupt-stats	Displays interrupt statistics.
show logging onboard module *slot*	Displays OBFL information for a specific module.

Command	Purpose
show logging onboard obfl-history	Displays history information.
show logging onboard obfl-logs	Displays log information.
show logging onboard stack-trace	Displays kernel stack trace information.
show logging onboard starttime	Displays OBFL logs from a specified start time.
show logging onboard status	Displays OBFL status information.

Use the **show logging onboard status** command to display the configuration status of OBFL.

```
switch# show logging onboard status
----------------------------
OBFL Status
----------------------------
Switch OBFL Log: Enabled

Module: 4 OBFL Log: Enabled
cpu-hog Enabled
credit-loss Enabled
environmental-history Enabled
error-stats Enabled
exception-log Enabled
interrupt-stats Enabled
mem-leak Enabled
miscellaneous-error Enabled
obfl-log (boot-uptime/device-version/obfl-history) Enabled
register-log Enabled
request-timeout Enabled
stack-trace Enabled
system-health Enabled
timeout-drops Enabled
stack-trace Enabled

Module: 22 OBFL Log: Enabled
cpu-hog Enabled
credit-loss Enabled
environmental-history Enabled
error-stats Enabled
exception-log Enabled
interrupt-stats Enabled
mem-leak Enabled
miscellaneous-error Enabled
obfl-log (boot-uptime/device-version/obfl-history) Enabled
register-log Enabled
request-timeout Enabled
stack-trace Enabled
system-health Enabled
timeout-drops Enabled
stack-trace Enabled
```

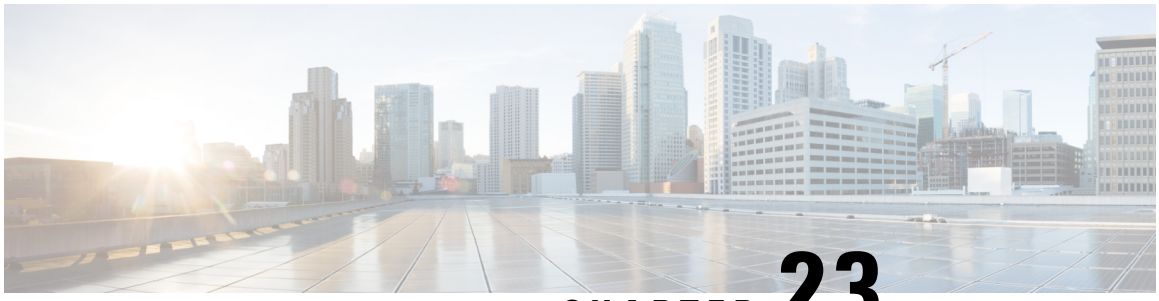
Use the **clear logging onboard** command to clear the OBFL information for each of the **show** command options listed.

Related Documents

This reference provides a related document for configuration files.

Table 27: Related Documents

Related Topic	Document Title
Configuration files	*Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide*



CHAPTER 23

Switched Port Analyzer

A switched port analyzer (SPAN) is a network montioring feature to analyze traffic between ports on Cisco NX-OS devices by directing the SPAN session traffic to a destination port with an external analyzer attached to it. You can define the sources and destinations to monitor in a SPAN session on the local device.

- Ports, on page 367
- Sessions, on page 369
- SPAN Functionality, on page 370
- Guidelines and Limitations, on page 370
- Configure SPAN Session, on page 383
- Configure UDF-Based SPAN, on page 386
- Configure SPAN Truncation, on page 389
- Configure SPAN for Multicast Tx Traffic Across Different LSE Slices, on page 391
- Configure SPAN to CPU, on page 393
- Shut Down or Resume SPAN Session, on page 395
- Verify SPAN Configuration, on page 397
- Configuration Examples, on page 397
- Related Documents, on page 398

Ports

Sources

The interfaces from which traffic can be monitored are called SPAN sources. Sources designate the traffic to monitor and whether to copy ingress (Rx), egress (Tx), or both directions of traffic.

SPAN sources include the following:

- Ethernet ports (but not subinterfaces)
- The inband interface to the control plane CPU

Note When you specify the supervisor inband interface as a SPAN source, the device monitors all packets that are sent by the Supervisor CPU.

- VLANs
  - When you specify a VLAN as a SPAN source, all supported interfaces in the VLAN are SPAN sources.
  - VLANs can be SPAN sources only in the ingress direction.
  - On supported Silicon One platforms, a VLAN source can be configured in RX, TX, or BOTH directions. On platforms where VLAN source direction is not enhanced, continue to follow the existing platform-specific VLAN-source limitations.
  - A single SPAN session can include mixed sources, such as interfaces, port channels, and VLAN sources, subject to platform scale and source-sharing restrictions.
- Satellite ports and host interface port channels on the Cisco Nexus 2000 Series Fabric Extender (FEX)
  - These interfaces are supported in Layer 2 access mode and Layer 2 trunk mode. They are not supported in Layer 3 mode, and Layer 3 subinterfaces are not supported.
  - Cisco Nexus 9300 and 9500 platform switches support FEX ports as SPAN sources in the ingress direction for all traffic and in the egress direction only for known Layer 2 unicast traffic flows through the switch and FEX. Routed traffic might not be seen on FEX HIF egress SPAN.

SPAN source ports have the following characteristics:

- A port configured as a source port cannot also be configured as a destination port.
- If you use the supervisor inband interface as a SPAN source, all packets generated by the supervisor hardware (egress) are monitored.

Note Rx is from the perspective of the ASIC (traffic egresses from the supervisor over the inband and is received by the ASIC/SPAN).

Note A single SPAN session can include mixed sources in any combination of the above.

Destinations

SPAN destinations refer to the interfaces that monitor source ports. Destination ports receive the copied traffic from SPAN sources. SPAN destinations include the following:

- Ethernet ports
- Port channels
- CPU as destination port
- Uplink ports on Cisco Nexus 9300 Series switches

Note FEX ports are not supported as SPAN destination ports.

SPAN destination ports have the following characteristics:

- A port configured as a destination port cannot also be configured as a source port.
- The same destination interface cannot be used for multiple SPAN sessions.
- Destination ports do not participate in any spanning tree instance. SPAN output includes bridge protocol data unit (BPDU) Spanning Tree Protocol hello packets.

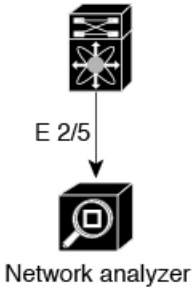
Sessions

You can create SPAN sessions to designate sources and destinations to monitor.

See the *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide* for information on the number of supported SPAN sessions.

This figure shows a SPAN configuration. Packets on three Ethernet ports are copied to destination port Ethernet 2/5. Only traffic in the direction specified is copied.

Figure 8: SPAN Configuration



Source Port	Direction	Destination Ports
E 2/1	Rx	E 2/5
E 2/2	Rx, Tx	
E 2/3	Tx	

503005

Localized SPAN Sessions

A SPAN session is localized when all of the source interfaces are on the same line card. A session destination interface can be on any line card.

Note A SPAN session with a VLAN source is not localized.

SPAN Functionality

SPAN Truncation

Beginning with Cisco NX-OS Release 7.0(3)I7(1), you can configure the truncation of source packets for each SPAN session based on the size of the MTU. Truncation helps to decrease SPAN bandwidth by reducing the size of monitored packets. Any SPAN packet that is larger than the configured MTU size is truncated to the given size. For example, if you configure the MTU as 300 bytes, the packets with greater than 300 bytes are truncated to 300 bytes.

SPAN truncation is disabled by default. To use truncation, you must enable it for each SPAN session.

ACL TCAM Regions

You can change the size of the ACL ternary content addressable memory (TCAM) regions in the hardware. For information on the TCAM regions used by SPAN sessions, see the Configuring IP ACLs chapter of the Cisco Nexus 9000 Series NX-OS Security Configuration Guide.

High Availability

The SPAN feature supports stateless and stateful restarts. After a reboot or supervisor switchover, the running configuration is applied. For more information on high availability, see the Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide

Default Settings

The following table lists the default settings for SPAN parameters.

Parameters	Default
SPAN sessions	Created in the shut state

Prerequisites

You must first configure the ports on each device to support the desired SPAN configuration. For more information, see the Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide.

Guidelines and Limitations

The following sections provide a comprehensive overview of the guidelines and limitations associated with configuring Switched Port Analyzer (SPAN) on Cisco Nexus 9000 Series devices. These guidelines cover important considerations such as session limits, hardware-based replication, source and destination port configurations, platform-specific restrictions, and behavior nuances related to ingress and egress traffic monitoring. Additionally, the limitations also address aspects like unsupported port types, filtering capabilities, and specific behaviors on certain line cards and platforms. For scale information and SPAN session limits, see the release-specific *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide.*

General Guidelines

- On Nexus 3000 Series switches running Cisco Nexus 9000 code, the Cisco Nexus 3232C and 3264Q switches do not support SPAN on CPU as destination.

- The show monitor session command displays incorrect statistics on the TX (egress) interface while mirroring traffic at line rate. This issue is seen in Cisco Nexus 93C64E-SG2-Q, Cisco Nexus 9364E-SG2-O switches with wide mode counters for polling statistics.
- Cisco N9336C-SE1 uses wide counters for statistics.
- A maximum of 48 source interfaces are supported per SPAN session (Rx and Tx, Rx, or Tx).
- All SPAN replication is performed in the hardware. The supervisor CPU is not involved.
- Traffic that is denied by an ACL may still reach the SPAN destination port because SPAN replication is performed on the ingress side prior to the ACL enforcement (ACL dropping traffic).
- Packets with FCS errors are not mirrored in a SPAN session.
- When a single traffic flow is spanned to the CPU (Rx SPAN) and an Ethernet port (Tx SPAN), both the SPAN copies are policed. Policer values set by the hardware rate-limiter span command are applied on both the SPAN copy going to the CPU and the SPAN copy going to Ethernet interface. This limitation applies to the following switches:
  - Cisco Nexus 9300- FX, FX2, FX3, GX platform switches
  - Cisco Nexus 9504, 9508, and 9516 platform switches with EX and FX line cards
- When a SPAN session contains source ports that are monitored in the transmit or transmit and receive direction, packets that these ports receive can be replicated to the SPAN destination port although the packets are not actually transmitted on the source ports. Some examples of this behavior on source ports are as follows:
  - Traffic that results from flooding
  - Broadcast and multicast traffic

General Limitations

- SPAN is supported in Layer 3 mode; however, SPAN is not supported on Layer 3 subinterfaces or Layer 3 port-channel subinterfaces.
- SPAN sessions cannot capture packets with broadcast or multicast MAC addresses that reach the supervisor, such as ARP requests and Open Shortest Path First (OSPF) protocol hello packets, if the source of the session is the supervisor Ethernet in-band interface. To capture these packets, you must use the physical interface as the source in the SPAN sessions.
- An access-group filter in a SPAN session must be configured as vlan-accessmap. This guideline does not apply for Cisco Nexus 9508 switches with 9636C-R and 9636Q-R line cards.
- Supervisor-generated stream of bytes module header (SOBMH) packets have all the information to go out on an interface and can bypass all forwarding lookups in the hardware, including SPAN and ERSPAN. CPU-generated frames for Layer 3 interfaces and the Bridge Protocol Data Unit (BPDU) class of packets are sent using SOBMH. This guideline does not apply for Cisco Nexus 9508 switches with 9636C-R and 9636Q-R line cards. The Cisco Nexus 9636C-R and 9636Q-R both support inband SPAN and local SPAN.
- Cisco NX-OS does not span Link Layer Discovery Protocol (LLDP) or Link Aggregation Control Protocol (LACP) packets when the source interface is not a host interface port channel.

- SPAN copies for multicast packets are made before rewrite. Therefore, the TTL, VLAN ID, any remarking due to an egress policy, and so on, are not captured in the SPAN copy.
- If SPAN is mirroring the traffic which ingresses on an interface in an ASIC instance and egresses on a Layer 3 interface (SPAN Source) on a different ASIC instance, then a Tx mirrored packet has a VLAN ID of 4095 on Cisco Nexus 9300 platform switches (except EX, FX, or FX2) and Cisco Nexus 9500 platform modular switches.
- When the trunk vlan and access VLAN are same, the tx mirrored packets will carry dot1q header. When the vlans are different, the packets don't carry dot1q header. This guideline does not apply for Cisco Nexus 9508 platform switches with 9636C-R and 9636Q-R line cards.
- The flows for post-routed unknown unicast flooded packets are in the SPAN session, even if the SPAN session is configured not to monitor the ports on which this flow is forwarded. This limitation applies to Network Forwarding Engine (NFE) and NFE2-enabled EOR switches and SPAN sessions that have Tx port sources.

Configuration Limitations

- You can configure maximum of 32 source VLANs while configuring SPAN session.
- You cannot configure a port as both a source and destination port.
- You can configure a SPAN session on the local device only

Release Specific Guidelines

- Beginning with Cisco NX-OS Release 10.1(2), SPAN is supported on the Cisco N9K-X9624D-R2 line card.
- Beginning with Cisco NX-OS Release 10.2(1q)F, SPAN is supported on the N9K-C9332D-GX2B platform switches.
- Beginning with Cisco NX-OS Release 10.2(2)F, Multicast SPAN Tx is supported on Cisco Nexus 9300-GX, 9300-GX2, and 9300-FX3 platform switches.
- Beginning with Cisco NX-OS Release 10.3(1)F, SPAN is supported on Cisco Nexus 9808 platform switches.
- Beginning with Cisco NX-OS Release 10.4(1)F, SPAN is supported on the following switches and line cards:
  - Cisco Nexus 9804 switch
  - Cisco Nexus 9332D-H2R switch
  - Cisco Nexus X98900CD-A and X9836DM-A line cards with Cisco Nexus 9808 and 9804 switches
- Beginning with Cisco NX-OS Release 10.4(2)F, Layer 3 port-channel interface as SPAN source is supported on 9808 and 9804 platform switches

  SPAN is supported on Cisco Nexus 93400LD-H1 platform switch.
- Beginning with Cisco NX-OS Release 10.5(3)F, SPAN is supported on Cisco Nexus 9364E-SG2 ToR switches.

Unsupported Features

- Configuring two SPAN or ERSPAN sessions on the same source interface with only one filter is not supported. If the same source is used in multiple SPAN or ERSPAN sessions, either all the sessions must have different filters or no sessions should have filters.
- SPAN mirroring is not supported for PBR traffic.
- Enabling UniDirectional Link Detection (UDLD) on the SPAN source and destination ports simultaneously is not supported. If UDLD frames are expected to be captured on the source port of such SPAN session, disable UDLD on the destination port of the SPAN session.
- SPAN is not supported for management ports.
- Statistics are not support for the filter access group.
- Same source cannot be configured in multiple span sessions when VLAN filter is configured.
- MTU truncation is not supported on Cisco Nexus 9504/9508 modular chassis with the N9K-X9636C-R, N9K-X9636Q-R, N9K-X9636C-RX, and N9K-X96136YC-R line cards.
- VLAN and ACL filters are not supported for FEX ports.

Guidelines for SPAN copies of access port dot1q headers

- When traffic ingresses from a trunk port or a routed port and egresses to an access port, an egress SPAN copy of an access port on a switch interface always has a dot1q header.
- When traffic ingresses from an access port and egresses to a trunk port or a routed port, an ingress SPAN copy of an access port on a switch interface does not have a dot1q header
- When traffic ingresses from an access port and egresses to an access port, an ingress/egress SPAN copy of an access port on a switch interface does not have a dot1q header.
- This behavior is applicable to Cisco Nexus 9300-EX, 9300-FX, 9300-FX2, 9300-FX3, 9300-GX, 9300-GX2, 9500 platform switches with N9K-X97160YC-EX, 9700-EX, 9700-FX, and 9700-GX line cards.

Guidelines related to VLAN

- VLAN SPAN monitors only the traffic that enters Layer 2 ports in the VLAN.
- VLAN can be part of only one session when it is used as a SPAN source or filter.
- VLAN ACL redirects to SPAN destination ports are not supported.
- When using a VLAN ACL to filter a SPAN, only **action forward** is supported; **action drop** and **action redirect** are not supported.
- The combination of VLAN source session and port source session is not supported. If the traffic stream matches the VLAN source session and port source session, two copies are needed at two destination ports. Due to the hardware limitation, only the VLAN source SPAN and the specific destination port receive the SPAN packets. This limitation applies only to the following Cisco devices:

Table 28: N9000 Series Switches

Cisco Nexus 93120TX	Cisco Nexus 93128TX	Cisco Nexus 9332PQ
Cisco Nexus 9372PX	Cisco Nexus 9372PX-E	Cisco Nexus 9372TX

Cisco Nexus 9396PX	Cisco Nexus 9372TX-E	Cisco Nexus 9396TX

Table 29: Cisco Nexus 9000 Series Line Cards, Fabric Modules, and GEM Modules

N9K-X9408PC-CFP2	N9K-X9536PQ	N9K-C9504-FM
N9K-X9432PQ	N9K-X9464TX	—

- When you filter a monitor session, make sure that the access-group specified must be a VACL, or VLAN access-map and not a regular ACL for filtering purpose. This guidelines is not applicable for Cisco Nexus 9508 switches with 9636C-R and 9636Q-R line cards.
- VLAN sources are spanned only in the Rx direction. This limitation does not apply to the following switch platforms which support VLAN spanning in both directions:
  - Cisco Nexus 9300-FX platform switches
  - Cisco Nexus 9300-FX2 platform switches
  - Cisco Nexus 9300-FX3 platform switches
  - Cisco Nexus 9300-GX platform switches
  - Cisco Nexus 9300-GX2 platform switches
  - Cisco Nexus 9300-H2R platform switches
  - Cisco Nexus 9504, 9508, and 9516 switches with the 97160YC-EX line card.
  - Cisco Nexus 9508 switches with 9636C-R and 9636Q-R line cards.
- If a VLAN source is configured as both directions in one session and the physical interface source is configured in two other sessions, Rx SPAN is not supported for the physical interface source session. This limitation applies to the Cisco Nexus 97160YC-EX line card.
- With regard to session filtering functionality, ACL filter is supported only in Rx source, and VLAN filter is supported in both Tx and Rx sources. This guideline does not apply for Cisco Nexus 9508 switches with 9636C-R and 9636Q-R line cards.

MTU Specific Guidelines

- Configuring MTU on a SPAN session truncates all packets egressing on the SPAN destination (for that session) to the MTU value specified.
  - The cyclic redundancy check (CRC) is recalculated for the truncated packet.
  - The bytes specified are retained starting from the header of the packets. The rest are truncated if the packet is longer than the MTU.
- When a packet exceeds the interface MTU, it is punted to software for fragmentation as the hardware cannot forward it. SPAN captures packets before this fragmentation, so the fragmented packets do not appear in SPAN captures.

SPAN/ERSPAN Specific Guidelines

- When configuring local SPAN sessions or ERSPAN-source monitor sessions with a filter access-group rule, we recommend to configure the necessary sub-commands of the VLAN access-map too, such as the match rule. For more information, see Configuration Example for a SPAN ACL.
- Truncation is supported only for local and ERSPAN source sessions. It is not supported for ERSPAN destination sessions.
- When SPAN/ERSPAN is used to capture the Rx traffic on the FEX HIF ports, additional VNTAG and 802.1Q tags are present in the captured traffic.

FEX Specific Guidelines

- The FEX NIF interfaces or port-channels cannot be used as a SPAN source or SPAN destination. If the FEX NIF interfaces or port-channels are specified as a SPAN source or SPAN destination, the software displays an unsupported error.
- If the sources used in bidirectional SPAN sessions are from the same FEX, the hardware resources are limited to two SPAN sessions.
- When sFlow is configured on N9K-C9508-FM-G with the N9K-X9716D-GX line card, disable sFlow before configuring SPAN sessions.

Cisco Nexus 9200 Platform Switches

The following guidelines and limitations apply only the Cisco Nexus 9200 platform switches. For scale information, see the release-specific Cisco Nexus 9000 Series NX-OS Verified Scalability Guide.

The following features are not supported:

- Rx SPAN for multicast without a forwarding interface on the same slice as the SPAN destination port.
- Tx SPAN for multicast, unknown multicast, and broadcast traffic
- Tx SPAN of CPU-generated packets
- Multiple ACL filters on the same source
- Use of ACL filter to span subinterface traffic on the parent interface
- When multiple egress ports on the same slice are congested by egressing SPAN traffic, those egress ports will not get the line rate.
- The CPU SPAN source can be added only for the Rx direction (SPAN packets coming from the CPU).
- SPAN packets to the CPU are rate limited and are dropped in the inband path. You can change the rate limit using the **hardware rate-limiter span** command. You can analyze SPAN copies on the supervisor using the **ethanalyzer local interface inband mirror detail** command.

The following features are supported:

- UDF-based SPAN
- VLAN Tx SPAN

Cisco Nexus 9300 Platform Switches

The following guidelines and limitations apply only the Cisco Nexus 9300 platform switches. For scale information, see the release-specific *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide.*

General Limitations :

- A SPAN copy of Cisco Nexus 9300 platform switch 40G uplink interfaces will miss the dot1q information when spanned in the Rx direction.



Note This limitation does not apply to Nexus 9300- FX/FX2 platform switches that have the 100G interfaces.

- When multiple egress ports on the same slice are congested by egressing SPAN traffic, those egress ports will not get the line rate on the Cisco Nexus 9300- FX/FX2 /FX3 /GX platform switches.
- Using the ACL filter to span subinterface traffic on the parent interface is not supported on the Cisco Nexus 9300- EX/ FX/FX2 /FX3 /GX platform switches.
- On the Cisco Nexus 9300-FX/FX2 /FX3 /GX platform switches, the CPU SPAN source can be added only for the Rx direction (SPAN packets coming from the CPU).
- On the Cisco Nexus 9300-FX/FX2 /FX3 /GX platform switches, SPAN packets to the CPU are rate limited and are dropped in the inband path. You can change the rate limit using the **hardware rate-limiter span** command. You can analyze SPAN copies on the supervisor using the **ethanalyzer local interface inband mirror detail** command.
- The following Cisco Nexus switches support sFlow and SPAN together:
  - Cisco Nexus 9336C-FX2
  - Cisco Nexus 93240YC-FX2
  - Cisco Nexus 93360YC-FX2

Not Supported Features :

- Cisco Nexus 9300-GX platform : SPAN does not support ECMP hashing/load balancing at the source.
- Cisco Nexus 9300 platform switches : Tx SPAN on 40G uplink ports



Note This limitation does not apply to Nexus 9300- FX/FX2 switches that have the 100G interfaces.

- Cisco Nexus 9200, 9300- FX/FXP/FX2/FX3/GX/GX2, 9300C, C9516-FM-E2, and C9508-FM-E2 switches : Tx SPAN of CPU-generated packets is not supported
- The Cisco Nexus 9300- FX/FX2 /FX3 /GX platform switches : Multiple ACL filters on the same source

Supported Features :

- Cisco Nexus 9300-FX2 platform switches : Simultaneous SPAN and sFlow functionality.

- Cisco Nexus 9300- FX/FX2 platform switches : Both NetFlow and SPAN can be enabled simultaneously, providing a viable alternative to using sFlow and SPAN.
- Cisco Nexus 9300-FX2 switches : Support sFlow and SPAN co-existence.
- Cisco Nexus 9300- FX platform switches : VLAN Tx SPAN is supported
- Cisco Nexus 9300 platform switches : Multiple ACL filters on the same source
- A single forwarding engine instance supports four SPAN sessions. For Cisco Nexus 9300 platform switches, if the first three sessions have bidirectional sources, the fourth session has hardware resources only for Rx sources.
- Cisco Nexus 9300- FX/FX2/FX3/FXP platform switches : FEX ports as SPAN sources only in the ingress direction.
- Cisco Nexus 9300 platform switches (excluding Cisco Nexus 9300- FX/FX2/FX3/FXP switches) : FEX ports as SPAN sources in the ingress direction for all traffic and in the egress direction only for known Layer 2 unicast traffic flows through the switch and FEX. Routed traffic might not be seen on FEX HIF egress SPAN.

Filtering limitations on egress (Tx) SPAN on all Cisco Nexus 9300- FX/FX2 /FX3 /GX **platform switches:**

- ACL filtering is not supported (applies to both unicast and Broadcast, Unknown Unicast and Multicast (BUM) traffic)
- VLAN filtering is supported, but only for unicast traffic
- VLAN filtering is not supported for BUM traffic

Release Based Support :

- Beginning with Cisco NX-OS Release 9.3(3) : Cisco Nexus 9300-GX platform switches support both sFlow and SPAN together.
- Beginning with Cisco NX-OS Release 9.3(5) : Cisco Nexus 9300-GX platform switches support SPAN truncation.
- Beginning with Cisco NX-OS Release 9.3(7) : sFlow and SPAN are supported on Cisco N9K-C93180YC-FX3 platform switches.
- Beginning from Cisco NX-OS Release 10.2(3)F : The FC span feature provides packet capture support for FC ports, SAN port channels, and VSANs for both NPV and SAN switching modes on Cisco Nexus C93180YC-FX, C9336C-FX2-E, and C93360YC-FX2 platform switches. FC ports, SAN Port channel, and VSANs as source are not supported in ERSPAN. FC ports, SAN Port channel, and VSANs cannot be added as source in more than one span sessions. The guideline—A single forwarding engine instance supports four active SPAN sessions—is also applicable to the FC span feature. SNMP support for FC span feature is not available.

SPAN Tx Feature Support

- For Tx interface SPAN with Layer 2 switch port and port-channel sources on Cisco Nexus 9300- FX/FX2 /FX3 /GX platform switches, only one copy is made per receiver unit regardless of how many Layer 2 members are receiving the stream in the same VLAN. For example, if e1/1-8 are all Tx direction SPAN sources and all are joined to the same group, the SPAN destination port sees one pre-rewrite copy of the stream, not eight copies. In addition, if for any reason one or more of those ports drops the packets on egress (for example, due to congestion), the packets may still reach the SPAN destination port. For the

Cisco Nexus 9732C-EX line card, one copy is made per unit that has members. For port-channel sources, the Layer 2 member that will SPAN is the first port-channel member.

- SPAN Tx broadcast and SPAN Tx multicast are supported for Layer 2 port and port-channel sources across slices on Cisco Nexus 9300- FX/FX2 /FX3 /GX platform switches, and the Cisco Nexus 9732C-EX line card, but only when IGMP snooping is disabled. (Otherwise, the slice limitation still applies.) These features are not supported for Layer 3 port sources, FEX ports (with unicast or multicast traffic), and VLAN sources.
- For SPAN Tx multicast for Layer 2, SPAN copies are created independent of multicast replication. Due to this, multicast and SPAN packet have different values for VLAN tag, which is the ingress interface VLAN ID.
- Configuring SPAN TX on same interface as sflow data-source port, the effective sflow process samples will be halved for that source port from the configured sampling rate.

UDF-based SPAN Feature Support

- UDF-based SPAN is supported on the Cisco Nexus 9300- FX/FX2 /FX3 /GX platform switches.
- UDF-SPAN acl-filtering only supports source interface rx. This limitation applies to the following switches:
  - Cisco Nexus 9332PQ
  - Cisco Nexus 9372PX
  - Cisco Nexus 9372PX-E
  - Cisco Nexus 9372TX
  - Cisco Nexus 9372TX-E
  - Cisco Nexus 93120TX

Cisco Nexus 9336C-SE1 switch

Beginning with 10.6(1)F, SPAN is supported on Cisco N9336C-SE1 switch. The guidelines and limitations include:

- A maximum of 10 active monitor (SPAN) sessions are supported at a time.
- Sharing of the same source port or interface across multiple sessions is not supported. SPAN mirrored packets use the default egress queue and do not have a dedicated SPAN egress queue.
- When a port-channel interface with multiple member ports is configured as a SPAN destination, only one member interface is used for mirrored traffic. Member selection is handled in software, which results in packet loss when membership changes.
- MTU truncation is supported only for 144 bytes in Rx mirroring and 80 bytes in Tx mirroring, excluding FCS.
- The features that are not supported include:
  - SPAN on subinterfaces,
  - sharing of the same source port or interface across sessions,

- tunnel ports,
- VLAN source,
- UDF, and
- ACL filter.

Cisco Nexus 9364E-SG2 switches

Beginning with 10.5(3)F, SPAN is supported on Cisco N9364E-SG2-O and N9364E-SG2-Q ToR switches. This section lists the guidelines and limitations that you need to follow.

- The switch supports a maximum of four active monitor sessions at a time, with session ID 4 reserved for SPAN on drop.
- N9364E-SG2 mirrors packets on sub-interface when parent interface is configured as source. SPAN mirrored packets do not have separate SPAN egress queue, they take the default queue.
- When multicast traffic is mirrored by local span, it is accounted as multicast under monitor port.
- When port-channel interface (with more than one member port) is configured as SPAN destination, only one member interface is used to send mirrored traffic. Member selection is done in software, which can lead to packet loss when membership changes.
- For drops on the SPAN destination (monitor) port, drop stats per interface per queue are not available.
- MTU truncation is supported for SPAN Rx mirroring. Beginning with Cisco NX-OS Release 10.6(1)F, MTU truncation is supported for local SPAN Tx mirroring. MTU truncation for SPAN supports 218 bytes excluding FCS. For Rx mirroring, packets are truncated to the configured 218 bytes and, for Tx mirroring, packets are truncated to 154 bytes excluding FCS.
- Only RX is supported on SPAN to CPU. However, beginning with Cisco NX-OS Release 10.6(1)F, Tx mirroring is also supported on SPAN to CPU .
- The features that are not supported include:
  - sharing of the same source port or interface across sessions,
  - tunnel ports,
  - VLAN source,
  - UDF, and
  - ACL filter.

N9300 Smart Switches

Beginning with NX-OS Release 10.6(2)F, SPAN is supported on N9324C-SE1U and N9348Y2C6D-SE1U ToR switches. This section lists the guidelines and limitations that you need to follow when you configure SPAN on this switch.

- **Sessions**

  —The switch supports a maximum of 10 active monitor sessions at a time, irrespective of the sessions being local SPAN or ERSPAN.

- **MTU truncation**

  —MTU truncation for SPAN supports 144 bytes. Rx mirrored packets are truncated to 144 bytes excluding FCS but Tx mirrored packets are truncated to 80 bytes, excluding FCS.

- —When multicast traffic on front panel is mirrored by local SPAN, it is accounted as multicast under monitor port.

- **Port-channel interface**

  —When port-channel interface with more than one member port is used as SPAN destination, only one member interface is used to send mirrored traffic. Member selection is done in software, which can lead to packet loss when membership changes.

- **Packet mirroring**

  —N9324C-SE1Uand N9348Y2C6D-SE1U mirrors packets on sub-interface when parent service-port-channel interface is configured as source. Beginning with NX-OS Release 10.6(2), mirrored packets are queued up in a dedicate SPAN Q on SPAN destination port. SPAN can be used to mirror traffic ingress or egress out of service-port-channel interface.

- SPAN to CPU is supported for both Rx and Tx mirroring.

- **Unsupported features**

  —The following features that are not supported.

  - mirroring packets on Layer 3 sub interfaces or Layer 3 port-channel sub interfaces when the respective parent interface is configured as source,
  - sharing of the same source port or interface across sessions,
  - tunnel ports, VLAN, SUP Ethernet, and management interface as a source, and
  - UDF and SPAN ACL filter.

Cisco Nexus 9500 Platform Switches

For scale information, see the release-specific *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*. The following guidelines and limitations apply only the Cisco Nexus 9500 platform switches:

Filtering Limitations on Egress (Tx) SPAN on 9500 platform switches with EX or FX line cards :

- ACL filtering is not supported (applies to both unicast and Broadcast, Unknown Unicast and Multicast (BUM) traffic)
- VLAN filtering is supported, but only for unicast traffic
- VLAN filtering is not supported for BUM traffic

Supported Features :

- For Cisco Nexus 9500 platform switches with EX or FX line cards, NetFlow and SPAN can both be enabled simultaneously, providing a viable alternative to using sFlow and SPAN.
- Cisco Nexus 9500 platform switches support VLAN Tx SPAN with the following line cards:
  - Cisco Nexus 97160YC-EX

- Cisco Nexus 9732C-EX
- Cisco Nexus 9732C-FX
- Cisco Nexus 9736C-EX
- Cisco Nexus 9736C-FX
- Cisco Nexus 9736Q-FX
- Cisco Nexus 9788TC-FX

- Cisco Nexus 9500 platform switches support multiple ACL filters on the same source.
- Cisco Nexus 9500 platform switches support FEX ports as SPAN sources in the ingress direction for all traffic and in the egress direction only for known Layer 2 unicast traffic flows through the switch and FEX. Routed traffic might not be seen on FEX HIF egress SPAN.
- Truncation is supported for Cisco Nexus 9500 platform switches with N9K-X97160YC-EX 9700-EX or 9700-FX line cards.
- On the Cisco Nexus 9500 platform switches, depending on the SPAN source's forwarding engine instance mappings, a single forwarding engine instance may support four SPAN sessions. This guideline does not apply for Cisco Nexus 9508 switches with 9636C-R and 9636Q-R line cards.
- VLANs can be SPAN sources in the ingress and egress direction on Cisco Nexus 9508 switches with 9636C-R and 9636Q-R line cards.

Not Supported Features :

- FEX and SPAN port-channel destinations are not supported on the Cisco Nexus 9500 platform switches with EX or FX line cards.
- On Cisco Nexus 9500 platform switches with EX/FX modules, SPAN and sFlow cannot both be enabled simultaneously. If one is active, the other cannot be enabled.
- Tx SPAN of CPU-generated packets is not supported on Cisco Nexus 9500 platform switches with EX-based line cards.
- TCAM carving is not required for SPAN/ERSPAN on the following line cards:
  - Cisco Nexus 9636C-R
  - Cisco Nexus 9636Q-R
  - Cisco Nexus 9636C-RX
  - Cisco Nexus 96136YC-R
  - Cisco Nexus 9624D-R2

Note All other switches supporting SPAN/ERSPAN must use TCAM carving.

- Same source interface cannot be configured in multiple SPAN sessions on N9K-X96136YC-R line card.
- Multiple ACL filters are not supported on the same source.

- SPAN does not support destinations on Cisco Nexus 9408PC-CFP2 line card ports.
- UDF-SPAN acl-filtering only supports source interface rx. This limitation applies to the following line cards:
  - Cisco Nexus 9564PX
  - Cisco Nexus 9464TX2
  - Cisco Nexus 9464TX
  - Cisco Nexus 9464TX2
  - Cisco Nexus 9564TX
  - Cisco Nexus 9464PX
  - Cisco Nexus 9536PQ
  - Cisco Nexus 9636PQ
  - Cisco Nexus 9432PQ

Cisco Nexus 9800 Platform Switches

For scale information, see the release-specific *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide* on Cisco.com. The following guidelines and limitations apply only the Cisco Nexus 9800 platform switches:

Supported Features :

- Only RX is supported on SPAN to CPU.
- A maximum of 10 monitor sessions are supported at a time.
- 10 active SPAN sessions are supported at a time.
- Beginning with Cisco NX-OS Release 10.6(1)F, SPAN is supported on Layer 2 ports.
- MTU truncation is only supported for 343 bytes on 9804 and 9808 switches excluding FCS .

General Limitations :

- SPAN mirrored packets do not have separate SPAN egress queue, they take the default queue.
- Due to SDK limitations, mirrored multicast traffic is processed as unicast traffic instead of multicast traffic.

Not Supported Features :

- Sharing of the same source port or interface across sessions is not supported.
- Monitor stats are not displayed for SPAN to CPU.
- SPAN is not supported on tunnel ports.
- VLAN as source is not supported on SPAN.
- MTU truncation is supported only on RX and not on TX.

- UDF is not supported.
- SPAN is not supported on subinterfaces.
- Port-channel interface as SPAN destination is not supported on 9804 and 9808 switches .

Configure SPAN Session

You can configure a SPAN session on the local device only. By default, SPAN sessions are created in the shut state. You can configure the CPU as the SPAN destination for the following platform switches:

- Cisco Nexus 9200 Series switches (beginning with Cisco NX-OS Release 7.0(3)I4(1))
- Cisco Nexus 9300-FX Series switches (beginning with Cisco NX-OS Release 7.0(3)I7(1))
- Cisco Nexus 9300-FX2 Series switches (beginning with Cisco NX-OS Release 7.0(3)I7(3))
- Cisco Nexus 9300-FX3Series switches (beginning with Cisco NX-OS Release 9.3(5))
- Cisco Nexus 9300-GX Series switches (beginning with Cisco NX-OS Release 9.3(3))
- Cisco Nexus 9500-EX Series switches with -EX/-FX line cards



Note Cisco NX-OS commands for this feature may differ from those in Cisco IOS. For bidirectional traditional sessions, you can configure the sessions without specifying the direction of the traffic.

Before you begin

You must configure the destination ports in access or trunk mode. For more information, see the Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide.

Procedure

Step 1 Enter the various modes to begin the configuration. Global configuration mode can be enabled using the command **configure terminal**. Enter the interface configuration mode on the selected slot and port using the command **interface***ethernet slot* / *port*

Step 2 Configure switchport parameters for the selected slot and port or range of ports using the command **switchport**. Configure the switchport interface as a SPAN destination using the command **switchport monitor**

You can repeat steps 1 and 2 to configure monitoring on additional SPAN destinations.

Step 3 Clear the configuration of the specified SPAN session using the command **no monitor session***session-number*

The new session configuration is added to the existing session configuration.

Step 4 You are now in the monitor configuration mode. Add the new session configuration to the existing session configuration. By default, a session is created in the shut state, and the session is a local SPAN session. The optional keyword shut specifies a shut state for the selected session. To add the new session configuration, enter the following command **monitor session***session-number* [**shut**]

Example:

```
                        switch(config)# monitor session 3

switch(config-monitor)#
```

Example:

```
switch(config)# monitor session 3 shut
switch(config-monitor)#
```

Step 5 Configures a description for the session using the command **description***description*

Example:

```
switch(config-monitor)# description my_span_session_3
```

By default, no description is defined. The description can be up to 32 alphanumeric characters.

Step 6 Configure sources and the traffic direction in which you need to copy the packets. You can enter a range of Ethernet ports, FC ports, a port channel, SAN port channels, an inband interface, a range of VLANs, a range of VSANs, or a satellite port or host interface port channel on the Cisco Nexus 2000 Series Fabric Extender (FEX) using the command **source** { **interface***type* [**rx** | **tx** | **both**] | [**vlan** { *number* | *range* }[**rx** | **tx** | **both**]} | [**vsan** { *number* | *range* }[**rx**]}

Example:

```
switch(config-monitor)# source interface ethernet 2/1-3, ethernet 3/1 rx
```

Example:

```
switch(config-monitor)# source interface fc1/1 both
```

Example:

```
switch(config-monitor)# source interface port-channel 2
```

Example:

```
switch(config-monitor)# source interface san-port-channel201 both
```

Example:

```
switch(config-monitor)# source interface sup-eth
0 rx
```

Example:

```
switch(config-monitor)# source vlan 3, 6-8 rx
```

```
switch(config)# monitor session <session-id>
switch(config-monitor)# source vlan <vlan-id | vlan-range> [rx | tx | both]
```

Example:

```
switch(config-monitor)# source vsan 500 rx
```

Example:

```
switch(config-monitor)# source interface ethernet 101/1/1-3
```

You can configure one or more sources, as either a series of comma-separated entries or a range of numbers. You can specify the traffic direction to copy as ingress (rx), egress (tx), or both. For a unidirectional session, the direction of the source must match the direction specified in the session. You can repeat this step to configure all SPAN sources.



Note Source VLANs are supported only in the ingress direction. Source FEX ports are supported in the ingress direction for all traffic and in the egress direction only for known Layer 2 unicast traffic.

Supervisor as a source is only supported in the Rx direction.

Source VSANs are also supported only in the ingress direction.

Step 7 Configure which VLANs to select from the configured sources using the command **filter vlan** { *number* | *range* }

Example:

```
switch(config-monitor)# filter vlan 3-5, 7
```

You can configure one or more VLANs, as either a series of comma-separated entries or a range of numbers. Repeat this step to configure all source VLANs to filter.



Note A FEX port that is configured as a SPAN source does not support VLAN filters.

Filters are not supported when the source is either FC interface or VSAN.

Step 8 Configure a destination for copied source packets using the command **destination interface***typeslot* / *port*. Enable the SPAN session using the command **no shut**

Example:

```
switch(config-monitor)# destination interface ethernet 2/5
```

Example:

```
switch(config-monitor)# no shut
```



Note FC ports are not supported as a destination interface. The SPAN destination port must be either an access port or a trunk port. You must enable monitor mode on the destination port. To configure the CPU as the SPAN destination , enter **sup-eth 0** for the interface type.

Step 9 Some of the commands that are optional are mentioned below

A few of the optional commands are :

- **filter access-group***acl-filter* used to associate an ACL with the SPAN session
- **show monitor session** { **all** | *session-number* | **range***session-range* } [**brief**] used to display the SPAN configurations
- **copy running-config startup-config** used to copy the running configuration to the startup configuration.

Example:

```
switch(config-monitor)# filter access-group ACL1
```



Note Filters are not supported when the source is either FC interface or VSAN.

Example:

```
switch# configure terminal
switch(config)# interface ethernet 2/5
switch(config-if)# switchport
switch(config-if)# switchport monitor
switch(config-if)# no shut
switch(config-if)# exit
switch(config)#
```

Example:

```
switch(config)# no monitor session 3
switch(config)# monitor session 3
switch(config-monitor)# source interface ethernet 2/1-3, ethernet 3/1 rx
switch(config-monitor)# source interface port-channel 2
switch(config-monitor)# source interface sup-eth 0 rx
switch(config-monitor)# source vlan 3, 6-8 rx
switch(config-monitor)# source interface ethernet 101/1/1-3
switch(config-monitor)# filter vlan 3-5, 7
switch(config-monitor)# destination interface ethernet 2/5
switch(config-monitor)# no shut
switch(config-monitor)# exit
switch(config)# show monitor session 3
switch(config)# copy running-config startup-config
```

Example:

```
switch(config)# monitor session 1
switch(config-monitor)# source interface fc 1/9/1
switch(config-monitor)# source interface san-port-channel 171
switch(config-monitor)# source vsan 3701
switch(config-monitor)# destination interface ethernet 1/8
switch(config-monitor)# no shutdown
switch(config-monitor)# exit
switch(config)# show monitor session 1
switch(config)# copy running-config startup-config
```

Example:

Configure a local SPAN session with a VLAN source in both directions

```
switch# configure terminal
switch(config)# monitor session 2
switch(config-monitor)# source vlan 300 both
switch(config-monitor)# destination interface ethernet 1/10
```

Configure UDF-Based SPAN

You can configure the device to match on user-defined fields (UDFs) of the outer or inner packet fields (header or payload) and to send the matching packets to the SPAN destination. Doing so can help you to analyze and isolate packet drops in the network.

Before you begin

Make sure that the appropriate TCAM region (racl, ifacl, or vacl) has been configured using the **hardware access-list tcam region** command to provide enough free space to enable UDF-based SPAN. For more information, see the "Configuring ACL TCAM Region Sizes" section in the *Cisco Nexus 9000 Series NX-OS Security Configuration Guide* .

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Define the UDF as follows: **udf***udf-name offset-base offset length*

Example:

```
switch(config)# udf udf-x packet-start 12 1
 switch(config)# udf udf-y header outer l3 20 2
```

- *udf-name* —Specifies the name of the UDF. You can enter up to 16 alphanumeric characters for the name.
- *offset-base* —Specifies the UDF offset base as follows, where **header** is the packet header to consider for the offset: **packet-start** | **header** { **outer** | **inner** { **l3** | **l4** }} .
- *offset* —Specifies the number of bytes offset from the offset base. To match the first byte from the offset base (Layer 3/Layer 4 header), configure the offset as 0.
- *length* —Specifies the number of bytes from the offset. Only 1 or 2 bytes are supported. To match additional bytes, you must define multiple UDFs.

You can define multiple UDFs, but Cisco recommends defining only required UDFs.

Step 3 Attach the UDFs to one of the following TCAM regions using the command **hardware access-list tcam region** { **racl | ifacl | vacl** } **qualify***qualifier-name*

Example:

```
switch(config)# hardware access-list tcam region
 racl qualify ing-l3-span-filter
```

- racl—Applies to Layer 3 ports.
- ifacl—Applies to Layer 2 ports
- vacl—Applies to source VLANs.

You can attach up to 8 UDFs to a TCAM region.

Note When the UDF qualifier is added, the TCAM region goes from single wide to double wide. Make sure enough free space is available; otherwise, this command will be rejected. If necessary, you can reduce the TCAM space from unused regions and then re-enter this command. For more information,see the "Configuring ACL TCAM Region Sizes" section in the Cisco Nexus 9000 Series NX-OS Security Configuration Guide.

Note The **no** form of this command detaches the UDFs from the TCAM region and returns the region to single wide.

Step 4 Save the change persistently through reboots and restarts by copying the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

Step 5 Reload the deviceusing the command **reload**

Example:

```
switch(config)# reload
```

Note Your UDF configuration is effective only after you enter **copy running-config startup-config** + **reload** .

Step 6 Create an IPv4 access control list (ACL) and enter the IP access list configuration mode using the command **ip access-list***span-acl*

Example:

```
switch(config)# ip access-list span-acl-udf-only
 switch(config-acl)#
```

Step 7 Enter one of the following commands to configure the ACL to match only on UDFs (example 1) or to match on UDFs along with the current access control entries (ACEs) for the outer packet fields (example 2).

- **permit udf***udf-name value mask*
- **permit ip***source destination***udf***udf-name value mask*

Example:

```
switch(config-acl)# permit udf udf-x 0x40 0xF0 udf-y 0x1001 0xF00F
```

Example:

```
switch(config-acl)# permit ip 10.0.0./24 any udf udf-x 0x02 0x0F udf-y 0x1001 0xF00F
```

A single ACL can have ACEs with and without UDFs together. Each ACE can have different UDF fields to match, or all ACEs can match for the same list of UDFs.

Step 8 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

This is an optional step.

Example:

This example shows how to configure UDF-based SPAN to match on the inner TCP flags of an encapsulated IP-in-IP packet using the following match criteria:

- Outer source IP address: 10.0.0.2
- Inner TCP flags: Urgent TCP flag is set
- Bytes: Eth Hdr (14) + Outer IP (20) + Inner IP (20) + Inner TCP (20, but TCP flags at 13th byte)

- Offset from packet-start: 14 + 20 + 20 + 13 = 67
- UDF match value: 0x20
- UDF mask: 0xFF

```
udf udf_tcpflags packet-start 67 1
hardware access-list tcam region racl qualify ing-l3-span-filter
copy running-config startup-config
reload
ip access-list acl-udf
permit ip 10.0.0.2/32 any udf udf_tcpflags 0x20 0xff
monitor session 1
source interface Ethernet 1/1
filter access-group acl-udf
```

This example shows how to configure UDF-based SPAN to match regular IP packets with a packet signature (DEADBEEF) at 6 bytes after a Layer 4 header start using the following match criteria:

- Outer source IP address: 10.0.0.2
- Inner TCP flags: Urgent TCP flag is set
- Bytes: Eth Hdr (14) + IP (20) + TCP (20) + Payload: 112233445566DEADBEEF7788
- Offset from Layer 4 header start: 20 + 6 = 26
- UDF match value: 0xDEADBEEF (split into two-byte chunks and two UDFs)
- UDF mask: 0xFFFFFFFF

```
udf udf_pktsig_msb header outer l4 26 2
udf udf_pktsig_lsb header outer l4 28 2
hardware access-list tcam region racl qualify ing-l3-span-filter
copy running-config startup-config
reload
ip access-list acl-udf-pktsig
permit udf udf_pktsig_msb 0xDEAD 0xFFFF udf udf_pktsig_lsb 0xBEEF 0xFFFF
monitor session 1
source interface Ethernet 1/1
filter access-group acl-udf-pktsig
```

Configure SPAN Truncation

You can configure truncation for local and SPAN source sessions only.

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Enter the monitor configuration mode for the specified SPAN session using the command **monitor session***session number*

Example:

```
switch(config)# monitor session 5
                        switch(config-monitor)#
```

Step 3 Configure the source interface using the command **source interface***typeslot* / *port* [**rx** | **tx** | **both**]

Example:

```
switch(config-monitor)# source interface ethernet 1/5 both
```

Step 4 Configure the MTU size for truncation using the command **mtu***size*

Example:

```
switch(config-monitor)# mtu 320
```

Example:

```
switch(config-monitor)# mtu ?
<320-1518> Enter the value of MTU truncation size for SPAN packets
```

Any SPAN packet that is larger than the configured MTU size is truncated to the configured size. The MTU ranges for SPAN packet truncation are:

- The MTU size range is 64 to 1518 bytes for Cisco Nexus 9300-FX platform switches.
-
- The MTU size is 343 bytes (excluding FCS) for Cisco Nexus 9808 and 9804 platform switches.

Step 5 Configure the Ethernet SPAN destination port using the command **destination interface***typeslot* / *port*

Example:

```
switch(config-monitor)# destination interface Ethernet 1/39
```

Step 6 Enable the SPAN session using the command **no shut**

Example:

```
switch(config-monitor)# no shut
```

By default, the session is created in the shut state.

Step 7 Display the SPAN configuration using the command **show monitor session***session*

Example:

```
switch(config-monitor)# show monitor session 5
```

This is an optional step.

Step 8 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config-monitor)# copy running-config startup-config
```

Example:

This example shows how to configure SPAN truncation for use with MPLS stripping:

```
mpls strip
ip access-list mpls
statistics per-entry
20 permit ip any any redirect Ethernet1/5
interface Ethernet1/5
switchport
switchport mode trunk
mtu 9216
no shutdown
monitor session 1
source interface Ethernet1/5 tx
mtu 64
destination interface Ethernet1/6
  no shut
```

Configure SPAN for Multicast Tx Traffic Across Different LSE Slices

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Configure SPAN for multicast Tx traffic across different leaf spine engine (LSE) slices using the command [**no**] **hardware multicast global-tx-span**

Example:

```
switch(config)# hardware multicast global-tx-span
```

Note Beginning from Cisco NX-OS Release 10.2(2)F, if source and destination are on different slices, use this command for multicast SPAN Tx.

Step 3 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

Step 4 Reload the device using the command **reload**

Example:

```
switch(config)# reload
```

Example:

Before Multicast Tx SPAN Is Configured

```
switch# show interface eth1/15-16, ethernet 1/27 counters

----------------------------------------
Port           InOctets    InUcastPkts
----------------------------------------
Eth1/15          580928              0
Eth1/16             239              0
Eth1/27               0              0

----------------------------------------
Port        InMcastPkts    InBcastPkts
----------------------------------------
Eth1/15            9077              0
Eth1/16               1              0
Eth1/27               0              0

----------------------------------------
Port          OutOctets   OutUcastPkts
----------------------------------------
Eth1/15             453              0
Eth1/16          581317              0
Eth1/27               0              0

----------------------------------------
Port       OutMcastPkts   OutBcastPkts
----------------------------------------
Eth1/15               4              0
Eth1/16            9080              0
Eth1/27               0              0
```

Configuring Multicast Tx SPAN

```
switch(config)# hardware multicast global-tx-span
Warning: Global Tx SPAN setting changed, please save config and reload
switch(config)# copy running-config start-up config
[########################################] 100%
Copy complete.
switch(config)# reload
This command will reboot the system. (y/n)?  [n] y
```

After Multicast Tx SPAN Is Configured

```
switch# show interface eth1/15-16, eth1/27 counters

----------------------------------------
Port           InOctets    InUcastPkts
----------------------------------------
Eth1/15          392576              0
Eth1/16               0              0
Eth1/27               0              0
----------------------------------------
Port        InMcastPkts    InBcastPkts
----------------------------------------
Eth1/15            6134              0
Eth1/16               0              0
Eth1/27               0              0

----------------------------------------
```

```
Port          OutOctets   OutUcastPkts
----------------------------------------
Eth1/15               0              0
Eth1/16          392644              0
Eth1/27          417112              0

----------------------------------------
Port       OutMcastPkts   OutBcastPkts
----------------------------------------
Eth1/15               0              0
Eth1/16            6135              0
Eth1/27            6134              0
```

Configure SPAN to CPU

A SPAN-to-CPU is for troubleshooting packet flow through Cisco Nexus 9000 Series switches. Similarly, to a normal SPAN or Encapsulated Remote SPAN (ERSPAN) session, a SPAN-to-CPU monitor session involves the definition of one or more source interfaces and traffic directions. Any traffic that matches the direction (TX, RX, or both) defined on a source interface is replicated to the supervisor CPU. This traffic is filtered and analyzed with the use of ethanalyzer or saved to a local storage device for reviewing the results.

To verify whether packets generated by the CPU of a Cisco Nexus 9000 Series Switches are transmitted out of a specific interface, Cisco recommends using a packet capture utility on the remote device connected to the interface.

1. **Configure SPAN as CPU destination**

   You must be able to configure CPU as monitor session destination and same must be configured on hardware. On Tahoe platforms, this configuration is supported for local span only as there is no customer requirement to support it for ERSPAN termination session. The same will be supported for N9K-C9508-FM-R2.

2. **Analyze SPAN Traffic**

   When SPAN traffic reaches mentioned supervisor CPU. The modules identify as SPAN packets and takes necessary actions and ethanalyzer displays these packets. The Ethanalyzer control plane packet capture utility can be used to view traffic replicated to the CPU. The mirror keyword in the Ethanalyzer command filters traffic such that only traffic replicated by a SPAN-to-CPU monitor session is shown. Ethanalyzer capture and display filters can be used to further limit the traffic displayed.

3. **Limit SPAN traffic rate**

   Spanned traffic for CPU must be rate limited to avoid control plane disruption. Ethanalyzer uses libpcap module for processing, stripping, and decoding packet headers. Ethanalyzer uses mirror option to display the span traffic reaching supervisor CPU. To match SPAN to CPU a separate span class is created. All the traffic will be created as SPAN class and separate rate is created for this class as Control Plane Policing (COPP). The COPP traffic rate limit will be 50 kbps.

4. **Filter ACL**

   This will give customers the ability to choose the traffic which they want to monitor. This feature will be supported on all kind of monitor session. For span to cpu this particularly important as traffic will be rate limited and so, it becomes important to categorize the traffic which is intended to be spanned.

Before you begin

SPAN-to-CPU has the following configuration guidelines and limitations:

- No ACL Filtering is supported on inband sources.
- Sources such as Physical Interfaces (L2 and L3), port channels, and L3 subinterface are supported with ACL filter.
- ACL Filter is supported for Rx sources only.
- No ACL filtering supported on VLAN sources.
- Configuring multiple span sessions for the same source is not supported.
- MTU truncation is not supported on N9K-X9636C-R, N9K-X9636Q-R, N9K-X9636C-RX, N9K-X96136YC-R, N9K-X9624D-R2, N9K-C9508-FM-R, N9K-C9504-FM-R, N9K-C9508-FM-R2, N9K-C9504-FM-R2, N3K-C36180YC-R, N3K-C3636C-R, and N3K-C36480LD-R2.
- ACL filters are not supported on N9K-X9624D-R2 Line card until Cisco NX-OS release 10.2(2)F.
- Beginning with Cisco NX-OS Release 10.2(3)F, ACL filters is supported on N9K-X9624D-R2 Line card.

Procedure

Step 1 Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Configure the CPU as the SPAN destination using the command **configure CPU as SPAN**

Example:

```
switch(config-monitor)# destination interface sup-eth0
```

Step 3 Configure the access list which will be honored for filtering using the command **configure ACL Filter**

Example:

```
switch(config-monitor)# filter access-group <acl_filter_name>
```

Step 4 Display spanned packets using the command **configure ethanalyzer**

Example:

```
switch# ethanalyzer local interface inband mirror
```

This example shows the output of monitor session.

```
show monitor session 1 session 1
type : local
state : up
acl-name : acl-name not specified
```

```
source intf :
rx : Eth3/44
tx : Eth3/44
both : Eth3/44
source VLANs :
rx :
tx :
both :
filter VLANs : filter not specified
source fwd drops :
destination ports : sup-eth0
PFC On Interfaces :
source VSANs :
rx :
```

This example shows the output of copp.

```
#
show policy-map interface control-plane | begin span
class-map copp-system-p-class-span (match-any)
match exception span
set cos 0
police cir 50 pps , bc 256 packets
module 1 : <Designated Module>
conformed 910228778 bytes;
7217965 packets;
violated 7217965 bytes;
0 packets;
module 3 :
conformed 0 bytes;
0 packets;
violated 0 bytes;
0 packets;
0 packets;
```

Shut Down or Resume SPAN Session

You can shut down SPAN sessions to discontinue the copying of packets from sources to destinations. You can shut down one session in order to free hardware resources to enable another session. By default, SPAN sessions are created in the shut state.

You can resume (enable) SPAN sessions to resume the copying of packets from sources to destinations. In order to enable a SPAN session that is already enabled but operationally down, you must first shut it down and then enable it.

You can configure the shut and enabled SPAN session states with either a global or monitor configuration mode command.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Shut down the specified SPAN sessions using the command [**no**] **monitor session** { *session-range* | **all** } **shut**

Example:

```
switch(config)# monitor session 3 shut
```

By default, sessions are created in the shut state.

The **no** form of the command resumes (enables) the specified SPAN sessions. By default, sessions are created in the shut state.



Note If a monitor session is enabled but its operational status is down, to enable the session, you must first specify the **monitor session shut** command followed by the **no monitor session shut** command.

Step 3 Enter the monitor configuration mode using the command **monitor session***session-number*

Example:

```
switch(config)# monitor session 3
 switch(config-monitor)#
```

The new session configuration is added to the existing session configuration.

Step 4 Shut down the SPAN session using the command [**no**] **shut**

Example:

```
switch(config-monitor)# shut
```

By default, the session is created in the shut state.

The **no** form of the command enables the SPAN session. By default, the session is created in the shut state.

Step 5 Display the status of SPAN sessions using the command **show monitor**

Example:

```
switch(config-monitor)# show monitor
```

This is an optional step.

Step 6 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

Verify SPAN Configuration

Display Commands

To display the SPAN configuration, perform one of the following tasks:

Command	Purpose
show monitor session { **all** | *session-number* | **range** *session-range* } [**brief**]	Displays the SPAN session configuration.

Configuration Examples

To configure a unidirectional SPAN session, configure the destination ports in access mode and enable SPAN monitoring as follows:

```
switch# configure terminal
switch(config)# interface ethernet 2/5
switch(config-if)# switchport
switch(config-if)# switchport monitor
switch(config-if)# no shut
switch(config-if)# exit
switch(config)#
```

Configure a SPAN session as follows:

```
switch(config)# no monitor session 3
switch(config)# monitor session 3
switch(config-monitor)# source interface ethernet 2/1-3, ethernet 3/1 rx
switch(config-monitor)# filter vlan 3-5, 7
switch(config-monitor)# destination interface ethernet 2/5
switch(config-monitor)# no shut
switch(config-monitor)# exit
switch(config)# show monitor session 3
switch(config)# copy running-config startup-config
```

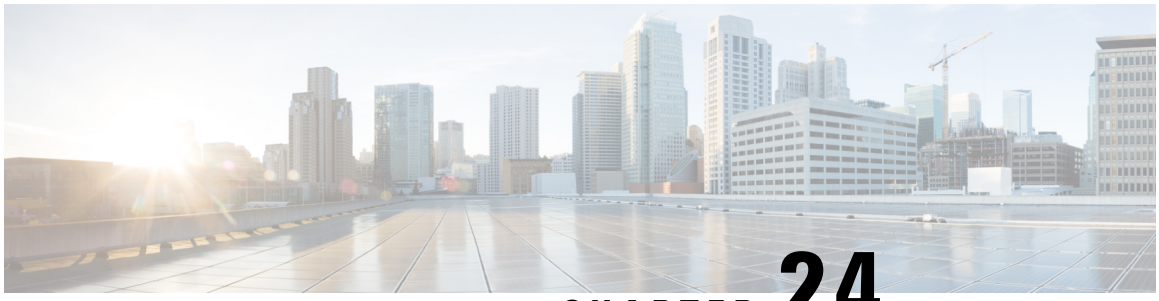
This example shows how to configure a SPAN ACL:

```
switch# configure terminal
switch(config)# ip access-list match_11_pkts
switch(config-acl)# permit ip 11.0.0.0 0.255.255.255 any
switch(config-acl)# exit
switch(config)# ip access-list match_12_pkts
switch(config-acl)# permit ip 12.0.0.0 0.255.255.255 any
switch(config-acl)# exit
switch(config)# vlan access-map span_filter 5
switch(config-access-map)# match ip address match_11_pkts
switch(config-access-map)# action forward
switch(config-access-map)# exit
switch(config)# vlan access-map span_filter 10
switch(config-access-map)# match ip address match_12_pkts
switch(config-access-map)# action forward
switch(config-access-map)# exit
switch(config)# monitor session 1
switch(config-erspan-src)# filter access_group span_filter
```

Related Documents

Table 30: Related Documents

Related Topic	Document Title
FEX	*Cisco Nexus 2000 Series NX-OS Fabric Extender Software Configuration Guide for Cisco Nexus 9000 Series Switches*



CHAPTER 24

Encapsulated Remote Switched Port Analyzer

An encapsulated remote switched port analyzer (ERSPAN) transports mirrored traffic over an IPv4 or IPv6 network, which provides remote monitoring of multiple switches across your network. The traffic is encapsulated at the source router and is transferred across the network. The packet is decapsulated at the destination router and then sent to the destination interface. Another method is that the destination can be the analyzer itself, which needs to understand the ERSPAN encapsulation format to parse the packet and access the inner (SPAN copy) frame.

- Overview, on page 399
- Guidelines and Limitations, on page 401
- Configure an ERSPAN Source Session, on page 408
- Configure an ERSPAN Destination Session, on page 412
- Configure an ERSPAN ACL, on page 414
- Configure UDF-Based ERSPAN, on page 417
- Configure ERSPAN Truncation, on page 420
- Configuration Examples, on page 422

Overview

Encapsulated Remote Switched Port Analyzer (ERSPAN) enables remote traffic monitoring on Cisco switches by designating sources, destinations, and session behaviors.

Sources

The interfaces from which traffic can be monitored are called ERSPAN sources. Sources designate the traffic to monitor and whether to copy ingress, egress, or both directions of traffic. ERSPAN sources include the following:

- Ethernet ports (but not subinterfaces)
- Port channels
- The inband interface to the control plane CPU

Note When you specify the supervisor inband interface as a SPAN source, the device monitors all packets that are sent by the Supervisor CPU.

Note If you use the supervisor inband interface as a SPAN source, all packets generated by the supervisor hardware (egress) are monitored.

Rx is from the perspective of the ASIC (traffic egresses from the supervisor over the inband and is received by the ASIC/SPAN).

- VLANs
  - When a VLAN is specified as an ERSPAN source, all supported interfaces in the VLAN are ERSPAN sources.
  - VLANs can be ERSPAN sources only in the ingress direction, except for Cisco Nexus 9300-FX/-FX2/-FX3/-GX series platform switches, and Cisco Nexus 9500 series platform switches with -EX/-FX line cards.
  - On supported Silicon One platforms, a VLAN source can be configured in RX, TX, or BOTH directions. Existing platform-specific restrictions continue to apply where the enhanced behavior is not supported.
  - An ERSPAN session with a VLAN source is not localized.

Note A single ERSPAN session can include mixed sources in any combination of the above.

Destinations

Destination ports receive the copied traffic from ERSPAN sources. The destination port is a port that is connected to the device such as a Remote Monitoring (RMON) probe or security device that can receive and analyze the copied packets from single or multiple source port. Destination ports do not participate in any spanning tree instance or any Layer 3 protocols

Cisco Nexus 9200, 9300-FX, and 9300-FX2 platform switches support an ERSPAN destination session configured on physical or port-channel interfaces in switchport mode through the use of GRE header traffic flow. The source IP address should be configured on the default VRF. Multiple ERSPAN destination sessions should be configured with the same source IP address.

Sessions

You can create ERSPAN sessions that designate sources to monitor. An ERSPAN session is localized when all of the source interfaces are on the same line card.

Note An ERSPAN session with a VLAN source is not localized

Truncation

Beginning with Cisco NX-OS Release 7.0(3)I7(1), you can configure the truncation of source packets for each ERSPAN session based on the size of the MTU. Truncation helps to decrease ERSPAN bandwidth by

reducing the size of monitored packets. Any ERSPAN packet that is larger than the configured MTU size is truncated to the given size. For ERSPAN, an additional ERSPAN header is added to the truncated packet from 54 to 166 bytes depending on the ERSPAN header type. For example, if you configure the MTU as 300 bytes, the packets are replicated with an ERSPAN header size from 354 to 466 bytes depending on the ERSPAN header type configuration.

ERSPAN truncation is disabled by default. To use truncation, you must enable it for each ERSPAN session.

Default Settings

The following table lists the default settings for ERSPAN parameters.

Table 31: Default ERSPAN Parameters

Parameters	Default
ERSPAN sessions	Created in the shut state

Guidelines and Limitations

Platform-specific guidelines and limitations apply, including support for jumbo frames, session limits, and feature restrictions. Some features, such as ERSPAN mirroring for PBR traffic, VLAN as source, and ERSPAN over VXLAN overlay, are not supported on certain platforms or configurations.

Prerequisites

You must first configure the ports on each device to support the desired ERSPAN configuration. For more information, see the Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide.

For ERSPAN session limits and scale information, see the release specific, *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide.*

Configuration Guidelines

- **ERSPAN destination handles jumbo frames for MTU differently based on the platform.**

  For the following Cisco Nexus 9300 platform switches and Cisco Nexus 9500 platform switches with supporting line cards, ERSPAN destination drops the jumbo frames.

  - 9300 Supported Line Cards - Cisco Nexus 9332PQ, Cisco Nexus 9372PX, Cisco Nexus 9372PX-E, Cisco Nexus 9372T, Cisco Nexus 9372TX-E, Cisco Nexus 93120TX
  - 9500 Supported Line Cards - Cisco Nexus 9564PX, Cisco Nexus 9464TX, Cisco Nexus 9464TX2, Cisco Nexus 9564TX, Cisco Nexus 9464PX, Cisco Nexus 9536PQ, Cisco Nexus 9636PQ, Cisco Nexus 9432PQ

  For Cisco Nexus 9200 platform switches and Cisco Nexus 9500 platform switches with supporting line cards, ERSPAN truncates the packets at port MTU and issues a TX Output error.

  - 9200 Supported Line Cards - Cisco Nexus 92160YC-X, Cisco Nexus 92304QC, Cisco Nexus 9272Q, Cisco Nexus 9232C, Cisco Nexus 9236C, Cisco Nexus 92300YC, Cisco Nexus 93108TC-EX, Cisco Nexus 93180LC-EX, Cisco Nexus 93180YC-EX

    - 9500 Supported Line Cards - Cisco Nexus 9736C-EX, Cisco Nexus 97160YC-EX, Cisco Nexus 9732C-EX, Cisco Nexus 9732C-EXM

- An access-group filter in an ERSPAN session must be configured as vlan-accessmap.
- Control plane packets that are generated by the Supervisor cannot be ERSPAN encapsulated or filtered by an ERSPAN access control list (ACL).
- A VLAN can be part of only one session when it is used as an ERSPAN source or filter.
- VLAN ERSPAN monitors only the traffic that leaves or enters Layer 2 ports in the VLAN.
- If you enable ERSPAN on a vPC and ERSPAN packets must be routed to the destination through the vPC, packets that come through the vPC peer link cannot be captured.
- ERSPAN copies for multicast packets are made before rewrite. Therefore, the TTL, VLAN ID, any remarking due to egress policy, and so on, are not captured in the ERSPAN copy.
- The timestamp granularity of ERSPAN Type III sessions is not configurable through the CLI. It is 100 picoseconds and driven through PTP.
- ERSPAN works on default and nondefault VRFs, but ERSPAN marker packets work only on the default VRF.

Unsupported Features

- Using the ACL filter to ERSPAN subinterface traffic on the parent interface is not supported on the Cisco Nexus 9200 platform switches and the Cisco Nexus 9300-FX/FX2/FX3/GX platform switches.
- ERSPAN mirroring is not supported for PBR traffic.
- ERSPAN with a Type three header is not supported in Cisco NX-OS Release 9.3(3).
- Statistics are not supported for the filter access group.
- ERSPAN is not supported for management ports.
- ERSPAN does not support destinations on Layer 3 port-channel subinterfaces.
- VLAN as a source is not supported with ERSPAN configuration on R-series linecards and N3K-C36180YC-R, N3KC36480LD-R2, and N3K-C3636C-R platform switches.
- ERSPAN is not supported over a VXLAN overlay.
- Configuring two SPAN or ERSPAN sessions on the same source interface with only one filter is not supported. If the same source is used in multiple SPAN or ERSPAN sessions either all the sessions must have different filters or no sessions should have filters. UDF-based ERSPAN is not supported.

Release Specification Limitations

Beginning with Cisco NX-OS Release 9.3(5)

- ERSPAN Type III Header is supported on Cisco Nexus 9300-GX platform switch.
- ERSPAN Destination Support is available on Cisco Nexus 9300-GX platform switch.

Beginning with Cisco NX-OS Release 10.1(2)

- ERSPAN is supported on the Cisco N9K-X9624D-R2 Line Card.
- IPv6 is supported on ERSPAN destination/termination on Cisco Nexus 9300-GX2, 9300-GX, 9300-FXP, 9300-FX2, 9300-EX, 9300-FX3, 9300-FX3S, and 9300-FX3P platform switches and N9K-X9716D-GX, N9K-X9736C-EX, N9K-X9732C-EX(X86_64 Atom), N9K-X9732C-EXM, N9K-X97160YC-EX, and N9K-X9736C-FX line cards.
- Only VRF default is supported on IPv6.
- You can only have one IPv6 address per switch.
- IPv6 is not supported with other tunnel features.
- You can bring up four ERSPAN destination sessions at a time on IPv6
- ERSPAN ID is unique per session and the range is 1–32 for IPv6.

Beginning with Cisco NX-OS Release 10.3(1)F

- ERSPAN is supported on Cisco Nexus 9808 platform switches.
- Only RX is supported on ERSPAN.
- Type 3 header is not supported on ERSPAN.
- ERSPAN destination/termination is not supported.

Beginning with Cisco NX-OS Release 10.4(1)F

- ERSPAN is supported on the Cisco Nexus 9332D-H2R switch.
- ERSPAN is supported on the Cisco Nexus 9804 switch but Type 3 header and ERSPAN destination/termination is not supported.
- ERSPAN is supported on the Cisco Nexus X98900CD-A and X9836DM-A line cards with Cisco Nexus 9808 and 9804 switches.

Beginning with Cisco NX-OS Release 10.4(2)F

- Cisco Nexus 9300-H2R platform switches supports SPAN on ACL drop in ingress direction for the ERSPAN source session.
- Layer 3 Port-channel interface as ERSPAN source and destination is supported on 9804 and 9808 platform switches. However, the following guidelines and limitations are applicable:
  - Load balancing of mirrored traffic on port channel is not supported.
  - Sharing of the same source port or interface across sessions is not supported.
  - A maximum of 10 monitor sessions are supported at a time.
  - 10 active ERSPAN sessions are supported at a time.
  - ERSPAN MTU truncation is only supported for 343 bytes on 9804 and 9808 switches excluding FCS.
  - ERSPAN Type 3 header is not supported.
  - ERSPAN destination/termination is not supported.

- ERSPAN Layer 2 interface (switch port) and VLAN as source is not supported. However, beginning with Cisco NX-OS Release 10.6(1)F, Layer 2 switch port (Ethernet/Port-channel) as source is supported.

- ERSPAN is supported on Cisco Nexus 93400LD-H1 platform switch.

Beginning with Cisco NX-OS Release 10.4(3)F

- ERSPAN is supported on Cisco Nexus 9364C-H1 platform switch.

Beginning with Cisco NX-OS Release 10.7(1)F

- ERSPAN VLAN source support is added for supported Silicon One platforms.
- A VLAN source can be configured in RX, TX, or BOTH directions where the platform supports the enhanced VLAN-source behavior. A maximum of 32 VLANs can be configured as sources in one ERSPAN source session.
- A source VLAN that is already configured in one monitor session cannot be configured as a source in another monitor session. A VLAN source can be configured with other supported source types in the same ERSPAN source session.
- If an interface or port channel that belongs to the source VLAN is also configured as a source in the same session, duplicate RX or TX copies can be received.
- Destination IP, VRF, ERSPAN ID, header type, TTL, DSCP, and marker-packet constraints are unchanged by this enhancement.
- Supported Silicon One platforms support a maximum of 63 source interfaces per ERSPAN session for Ethernet and port-channel sources. The 63 source-interface limit applies per session for RX and TX, RX, or TX direction configuration. Configurations with 64 or more Ethernet sources or 64 or more port-channel sources are rejected.
- Existing ERSPAN source-session CLI syntax is unchanged. No changes are introduced for ERSPAN encapsulation behavior, header type, marker packets, MTU truncation, session direction, or destination behavior.
- Downgrading a configuration with more than 48 source interfaces to a release that does not support this enhancement may require manual adjustment.

Egress (Tx) ERSPAN Guidelines

- The flows for post-routed unknown unicast flooded packets are in the ERSPAN session, even if the ERSPAN session is configured to not monitor the ports on which this flow is forwarded. This limitation applies to Network Forwarding Engine (NFE) and NFE2-enabled EOR switches and ERSPAN sessions that have TX port sources.
- For ERSPAN Tx multicast for Layer 2, ERSPAN copies are created independent of multicast replication. Due to this, multicast and SPAN packet have different values for VLAN tag, which is the ingress interface VLAN ID.
- The following guidelines and limitations apply to ingress (Rx) ERSPAN:
  - VLAN sources are spanned only in the Rx direction.
  - Session filtering functionality (VLAN or ACL filters) is supported only for Rx sources.

    - VLANs are supported as ERSPAN sources only in the ingress direction.

- Priority flow control (PFC) ERSPAN has the following guidelines and limitations:
    - It cannot coexist with filters.
    - It is supported only in the Rx direction on physical or port-channel interfaces. It is not supported in the Rx direction on VLAN interfaces or in the Tx direction.

- The following guidelines and limitations apply to FEX ports:
    - If the sources used in bidirectional ERSPAN sessions are from the same FEX, the hardware resources are limited to two ERSPAN sessions.
    - FEX ports are supported as ERSPAN sources in the ingress direction for all traffic and in the egress direction only for known Layer 2 unicast traffic.
    - Cisco Nexus 9300 platform switches do not support ERSPAN destination being connected on a FEX interface. The ERSPAN destination must be connected to a front panel port.
    - VLAN and ACL filters are not supported for FEX ports. It cannot coexist with filters.

- The following guidelines and limitations apply to ERSPAN destination:
    - Cisco Nexus 9200, 9300-EX, 9300-FX, and 9300-FX2 platform switches support an ERSPAN destination session that is configured on physical or port-channel interfaces in switchport mode by using GRE header traffic flow.
    - ERSPAN destination cannot coexist with other tunnel features such as MPLS and VXLAN for Cisco Nexus 9200, 9300, 9300-EX, 9300-FX, and 9300-FX2 platform switches.
    - On Cisco Nexus 9300-GX switches, dot1q-tagged broadcast or multicast packets passing through a device where the ERSPAN destination session is active, get tagged with the native VLAN instead of the correct VLAN due to a hardware limitation.
    - ERSPAN destination supports only default VRF.
    - Cisco Nexus 9300-EX/FX switches cannot serve as an ERSPAN destination for Cisco Nexus 3000 and non-EX/FX Cisco Nexus 9000 switches.

ERSPAN over IPv6 Guidelines

-
- This feature is not supported for Load balancing across egress port-channel members and egress ECMP path.
- This feature is not supported for header-type 3, udf in filter ACL, and marker-packets.
- This feature is not supported for FEX host interface as ERSPAN source with IPv6.

Platform-Specific Limitations

- A maximum of 48 source interfaces are supported per ERSPAN session (Rx and Tx, Rx, or Tx).

- The number of ERSPAN sessions per line card reduces to two if the same interface is configured as a bidirectional source in more than one session.
- Packets with FCS errors are not mirrored in an ERSPAN session.
- TCAM carving is not required for SPAN/ERSPAN on the following line cards:
  - Cisco Nexus 9636C-R
  - Cisco Nexus 9636Q-R
  - Cisco Nexus 9636C-RX
  - Cisco Nexus 96136YC-R
  - Cisco Nexus 9624D-R2

Note All other switches supporting SPAN/ERSPAN must use TCAM carving.

- The same source can be part of multiple sessions.
- ERSPAN mirrored packets do not have separate SPAN egress queue, they take the default queue.
- When port-channel interface (with more than one member port) is configured as ERSPAN destination, only one member interface is used for sending out mirrored traffic.
- The member selection is done in software, so there will be packet loss when membership changes.
- When configuring local SPAN sessions or ERSPAN-source monitor sessions with a filter access-group rule, we recommend to configure the necessary sub-commands of the VLAN access-map too, such as the match rule. For more information, see Configuration Example for an ERSPAN ACL.

ERSPAN Guidelines and Limitations for Cisco Nexus 9364E-SG2 Switches

Beginning with Cisco NX-OS Release 10.5(3)F, ERSPAN is supported on Cisco N9364E-SG2-O and N9364E-SG2-Q ToR switches. This section lists the guidelines and limitations that you need to follow.

- **Sessions**

  —The switch supports a maximum of four active monitor sessions at a time, with session ID 4 reserved for SPAN on drop.

- **MTU truncation**

  — . MTU truncation is supported for ERSPAN Rx mirroring. Beginning with Cisco NX-OS Release 10.6(1)F, MTU truncation is supported for ERSPAN Tx mirroring. MTU truncation for ERSPAN supports 218 bytes excluding FCS. For Rx mirroring, packets are truncated to the configured 218 bytes and, for Tx mirroring, packets are truncated to 154 bytes excluding FCS. ERSPAN encapsulation is done on the truncated packets.

- **Port-channel interface**

  —When port-channel interface with more than one member port is configured as ERSPAN destination, only one member interface is used to send mirrored traffic. Member selection is done in software, which can lead to packet loss when membership changes.

- **Drop stats**

  —For drops on the ERSPAN destination interface, drop stats per interface per queue are not available.

- **Packet mirroring**

  —In ERSPAN, multicast mirrored packet is accounted as unicast under ERSPAN destination interface. ERSPAN mirrored packets do not have separate SPAN egress queue, they take the default queue.

- **Unsupported features**

  —The features that are not supported include:

  - load balancing of mirrored traffic on port channel,
  - sharing of the same source port or interface across sessions,
  - ERSPAN Type 3 header,
  - ERSPAN destination/termination,
  - VLAN as source, and
  - ACL filter.

ERSPAN Guidelines and Limitations for N9300 Smart and Nexus 9324C-SE1U Switches

Beginning with NX-OS Release 10. 6(2)F , ERSPAN is supported on N9324C-SE1U and N9348Y2C6D-SE1U ToR switches. This section lists the guidelines and limitations that you need to follow while configuring ERSPAN on this switch.

- **Sessions**

  —The switch supports a maximum of 10 active monitor sessions at a time, irrespective of the sessions being local SPAN or ERSPAN.

- **MTU truncation**

  —MTU truncation for ERSPAN supports 144 bytes. Rx mirrored packets are truncated to 144 bytes excluding FCS but Tx mirrored packets are truncated to 80 bytes, excluding FCS.

- **Port-channel interface**

  —When port-channel interface with more than one member port is used as ERSPAN destination, only one member interface is used to send mirrored traffic. Member selection is done in software, which can lead to packet loss when membership changes.

- **Packet mirroring**

  —N9324C-SE1U and N9348Y2C6D-SE1U mirrors packets on sub-interface when parent service-port-channel interface is configured as source. Beginning with NX-OS Release 10.6(2), mirrored packets are queued up in a dedicate SPAN Q on ESPRAN destination port.ERSPAN can be used to mirror traffic ingress or egress out of service- port-channel interface.

- **Unsupported features**

  —The features that are not supported include:

  - mirroring packets on Layer 3 sub interfaces or Layer 3 port-channel sub interfaces when the respective parent interface is configured as source,

- sharing of the same source port or interface across sessions,
- tunnel ports, VLAN, SUP Ethernet, and management interface as a source,
- ERSPAN Type 3 header,
- ERSPAN destination/termination, and
- UDF and ERSPAN ACL filter.

Configure an ERSPAN Source Session

Note Be aware that the Cisco NX-OS commands for this feature may differ from those commands used in Cisco IOS.

You can configure an ERSPAN session on the local device only. By default, ERSPAN sessions are created in the shut state.

Note ERSPAN does not monitor any packets that are generated by the supervisor, regardless of their source.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Configure the ERSPAN global origin IP v4 or IPv6 address using the command **monitor erspan origin ip-address** *ip-address* **global** or **monitor erspan origin ipv6-address** *ipv6-address* **global**

Example:

```
switch(config)# monitor erspan origin ip-address 10.0.0.1 global
switch(config)# monitor erspan origin ipv6-address 2001:DB8:1::1 global
```

Step 3 Clear the configuration of the specified ERSPAN session using the command **no monitor session** { *session-number* | **all** }

Example:

```
switch(config)# no monitor session 3
```

The new session configuration is added to the existing session configuration.

Step 4 Configure an ERSPAN Type II source session using the command **monitor session** { *session-number* | **all** } **type erspan-source** [**shut**] and configure a description for the session using the command **description** *description* .

Example:

```
switch(config)# monitor session 3 type erspan-source
switch(config-erspan-src)#
```

Example:

```
switch(config-erspan-src)# description erspan_src_session_3
```

By default the session is bidirectional. The optional keyword shut specifies a shut state for the selected session. By default, no description is defined. The description can be up to 32 alphanumeric characters.

Step 5 Configure the sources and traffic direction in which to copy packets using the command **source** { **interface** *type* [**tx** | **rx** | **both**] **vlan** { **number** | **range** } [**tx** | **rx** | **both**] }

Example:

```
switch(config-erspan-src)# source interface
ethernet 2/1-3, ethernet 3/1 rx
```

Example:

```
switch(config-erspan-src)# source interface
port-channel 2
```

Example:

```
switch(config-erspan-src)# source interface
sup-eth 0 rx
```

Example:

```
switch(config-erspan-src)# source vlan 3, 6-8 rx
```

Example:

```
switch(config-erspan-src)# source interface
ethernet 101/1/1-3
```

You can enter a range of Ethernet ports, a port channel, an inband interface, a range of VLANs, or a satellite port or host interface port channel on the Cisco Nexus 2000 Series Fabric Extender (FEX). You can configure one or more sources, as either a series of comma-separated entries or a range of numbers. You can specify the traffic direction to copy as ingress, egress, or both. For a unidirectional session, the direction of the source must match the direction specified in the session.

Repeat this step to configure all ERSPAN sources.



Note Source VLANs are supported only in the ingress direction. Source FEX ports are supported in the ingress direction for all traffic and in the egress direction only for known Layer 2 unicast traffic. Supervisor as a source is only supported in the Rx direction.

Step 6 Configure which VLANs to select from the configured sources using the command **filter vlan** { *number* | *range* }

Example:

```
switch(config-erspan-src)# filter vlan 3-5, 7
```

You can configure one or more VLANs, as either a series of comma-separated entries or a range of numbers. For information on the VLAN range, see the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide.

Repeat this step to configure all source VLANs and to filter.

Note A FEX port that is configured as an ERSPAN source does not support VLAN filters.

Step 7 Associate an ACL with the ERSPAN session using the command **filter access-group** *acl-filter*

Example:

```
switch(config-erspan-src)# filter access-group
ACL1
```

This is an optional step. You can create an ACL using the standard ACL configuration process. For more information, see the Cisco Nexus 9000 Series NX-OS Security Configuration Guide.

Note Before executing this command, configure ip access list and associated vlan access map . See Configuring ERSPAN ACL .

Step 8 Follow the following commands to configure the various values for the ERSPAN session.

- Configure the destination IP v4 or IPv6 address in the ERSPAN session (Mandatory) - **destination ip** *ip-address*
- Configure the ERSPAN ID for the ERSPAN source session (Mandatory) - **erspan-id** *erspan-id*
- Configure the virtual routing and forwarding (VRF) instance that the ERSPAN source session uses for traffic forwarding (Mandatory) - **vrf** *vrf-name*
- Configure the IP time-to-live (TTL) value for the ERSPAN traffic (Optional) - **ip ttl** *ttl-number*
- Configure the differentiated services code point (DSCP) value of the packets in the ERSPAN traffic (Optional) - **ip dscp** *dscp-number*
- Enable the ERSPAN marker packet for a session in order to recover the real value of the ERSPAN timestamp (Optional) - [**no**] **marker-packet** *milliseconds*

Example:

```
switch(config-erspan-src)# destination ip 10.1.1.1
```

```
switch(config-erspan-src)# destination ipv6 2001:DB8:1::1
```

Example:

```
switch(config-erspan-src)# erspan-id 5
```

Example:

```
switch(config-erspan-src)# vrf default
```

Example:

```
switch(config-erspan-src)# ip ttl 25
```

Example:

```
switch(config-erspan-src)# ip dscp 42
```

Example:

```
switch(config-erspan-src)# marker-packet 100
```

Note Only one destination IP v4 or IPv6 address is supported per ERSPAN source session. The ERSPAN ID range is from 1 to 1023. The VRF name can be any case-sensitive, alphanumeric string up to 32 characters. The range for IP TTL is from 1 to 255. The range for IP DSCP is from 0 to 63. The interval for marker packet can range from 100 to 1000 milliseconds. The **no** form of this command disables the marker packet for the session. ERSPAN marker packets only apply to Type 3 sessions.

Step 9 Enable the ERSPAN source session using the command **no shut** and exit the monitor configuration mode using the command **exit**

Example:

```
switch(config-erspan-src)# no shut
```

Example:

```
switch(config-erspan-src)# exit
switch(config)#
```

By default, the session is created in the shut state.

This example shows how to configure an ERSPAN source session over IPv6:

```
switch# configure terminal
switch(config)# monitor erspan origin ipv6-address 2001::10:0:0:9 global
switch(config)# moni session 10 type erspan-source
switch(config-erspan-src)# erspan-id 10
switch(config-erspan-src)# vrf default
switch(config-erspan-src)# source interface ethernet 1/64
switch(config-erspan-src)# destination ipv6 2001:DB8:1::1
```

Configure an ERSPAN source session with a VLAN source in both directions

```
switch# configure terminal
switch(config)# monitor erspan origin ip-address 10.0.0.1 global
switch(config)# monitor session 3 type erspan-source
switch(config-erspan-src)# erspan-id 3
switch(config-erspan-src)# vrf default
switch(config-erspan-src)# destination ip 10.1.1.2
switch(config-erspan-src)# source vlan 300 both
```

What to do next

Verify ERSPAN Source Session

To display the ERSPAN configuration, perform one of the following tasks:

Command	Purpose
show monitor session {**all** | *session-number* | **range** *session-range*} [**brief**]	Displays the ERSPAN session configuration.
show running-config monitor	Displays the running ERSPAN configuration.
show startup-config monitor	Displays the ERSPAN startup configuration.

Configure an ERSPAN Destination Session

Use this task to configure an ERSPAN destination session, which allows you to copy packets from a source IP address to destination ports on the local device.

You can configure an ERSPAN destination session to copy packets from a source IP address to destination ports on the local device. By default, ERSPAN destination sessions are created in the shut state.

Before you begin

Ensure that you have already configured the destination ports in switchport monitor mode.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal** and enter interface configuration mode on the selected slot and port or range of ports using the command **interface ethernet** *slot/port* [*-port*]

Example:

```
switch# configure terminal
switch(config)#
```

Example:

```
switch(config)# interface ethernet 2/5
switch(config-if)#
```

Step 2 Configure the switchport parameters for the selected slot and port or range of ports using the command **switchport** and configure the switchport modes for the selected slot and port or range of ports using the command **switchport mode access** | **trunk**

Example:

```
switch(config-if)# switchport
```

Example:

```
switch(config-if)# switchport mode trunk
```

Step 3 Configure the switchport interface as an ERSPAN destination using the command **switchport monitor** and repeat steps 1 to 3 to configure monitoring on additional ERSPAN destinations.

Example:

```
switch(config-if)# switchport monitor
```

Step 4 Clear the configuration of the specified ERSPAN session using the command **no monitor session** { *session-number* | **all** } and configure an ERSPAN destination session using the command **type erspan-destination**

Example:

```
switch(config-if)# no monitor session 3
```

Example:

```
switch(config-if)# monitor session 3 type erspan-destination
switch(config-erspan-dst)#
```

The new session configuration is added to the existing session configuration.

Step 5 Configure a description for the session using the command **description** *description*

Example:

```
switch(config-erspan-dst)# description erspan_dst_session_3
```

By default, no description is defined. The description can be up to 32 alphanumeric characters.

Step 6 Configure the source IPv4 or IPv6 address in the ERSPAN session using the command **source ip** *ip-address*

source ipv6 *ipv6-address*

Example:

```
switch(config-erspan-dst)# source ip 10.1.1.1
switch(config-erspan-dst)# source ipv6 2001:DB8:1::1
```

The source IPv4 or IPv6 address is a locally configured IPv4 or IPv6 address. The source IPv4 or IPv6 address in an ERSPAN destination session must match the destination IPv4 or IPv6 address configured in the ERSPAN source session from which the encapsulated data is received. Only one source IPv4 or IPv6 address is supported per ERSPAN destination session.



Note IPv6 is supported from Cisco NX-OS Release 10.2(3)F.

Step 7 Configure a destination for copied source packets using the command **destination** {[**interface** [*type slot/port* [*-port*]]] [**port-channel** *channel-number*]]} . Repeat the step to configure all the ERSPAN destinations.

Example:

```
switch(config-erspan-dst)# destination interface ethernet 2/5
```

You can configure a destination interface.



Note You can configure destination ports as trunk ports.

Step 8 Configure the ERSPAN ID for the ERSPAN session using the command **erspan-id** *erspan-id* and enable the ERSPAN destination session using the command **no shut** .

Example:

```
switch(config-erspan-dst)# erspan-id 5
```

Example:

```
switch(config-erspan-dst)# no shut
```

The range for the ERSPAN ID is from 1 to 1023. By default, the destination session is created in the shut state.

Step 9 Exit monitor configuration mode and exit the global configuration mode using the command **exit**

Example:

```
switch(config-erspan-dst)# exit
```

Example:

```
switch(config)# exit
```

This example shows how to configure an ERSPAN destination session over IPv4:

The **destination interface eth1/1** is in switchport monitor mode. This interface can not co-exist with mpls strip, tunnel, nv overlay, vn-segment-vlan-based, mpls segment-routing, mpls evpn, mpls static, mpls oam, mpls l3vpn , mpls ldp, and nv overlay evpn features.

```
switch# monitor session 1 type erspan-destination
switch(config)# erspan-id 1
switch(config-erspan-dst)# source ip 10.1.1.1
switch(config-erspan-dst)# destination interface eth1/1
switch(config-erspan-dst)# no shut
switch(config-erspan-dst)# exit
```

This example shows how to configure an ERSPAN destination session over IPv6:

The **destination interface eth1/1** is in switchport monitor mode. This interface can not co-exist with mpls strip, tunnel, nv overlay, vn-segment-vlan-based, mpls segment-routing, mpls evpn, mpls static, mpls oam, mpls l3vpn , mpls ldp, and nv overlay evpn features.

```
switch# monitor session 1 type erspan-destination
switch(config)# erspan-id 1
switch(config-erspan-dst)# source ipv6 2001:DB8:1::1
switch(config-erspan-dst)# destination interface eth1/1
switch(config-erspan-dst)# no shut
switch(config-erspan-dst)# exit
```

What to do next

Verify ERSPAN Destination Session

Use the following optional commands to verify the destination session:

- Display the ERSPAN session configuration using the command **show monitor session** { **all** | *session-number* | **range** *session-range* }
- Display the running ERSPAN configuration using the command **show running-config monitor**
- Display the ERSPAN startup configuration using the command **show startup-config monitor**
- Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Configure an ERSPAN ACL

You can create an IPv4 ERSPAN ACL on the device and add rules to it.

To modify the DSCP value or the GRE protocol, you need to allocate a new destination monitor session. A maximum of four destination monitor sessions are supported.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Create the ERSPAN ACL and enter IP ACL configuration mode using the command **ip access-list** *acl-name*

Example:

```
switch(config)# ip access-list erspan-acl
switch(config-acl)#
```

The *acl-name* argument can be up to 64 characters.

Step 3 Create a rule in the ERSPAN ACL using the command [*sequence-number*] { **permit** | **deny** } *protocol source destination*

Example:

```
switch(config-acl)# permit ip 192.168.2.0/24 any
```

Example:

```
switch(config)# ip access-list match_11_pkts
switch(config-acl)# permit ip 10.0.0.0/24 any
switch(config-acl)# exit
```

You can create many rules.

The *sequence-number* argument can be a whole number between 1 and 4294967295. The **permit** and **deny** commands support many ways of identifying traffic.

Step 4 Enter VLAN access-map configuration mode for the VLAN access map specified using the command **vlan access-map erpsan-acl** *map name* [*sequence-number*]

Example:

```
switch(config)# vlan access-map erspan_filter
```

If the VLAN access map does not exist, the device creates it. If you do not specify a sequence number, the device creates a new entry whose sequence number is 10 greater than the last sequence number in the access map.

Step 5 Specify an ACL for the access-map entry using the command **match ip address** *acl-name* and specify the action that the device applies to traffic that matches the ACL using the command **action forward** .

Example:

```
switch(config-access-map)# match ip address erspan-acl
```

Example:

```
switch(config-access-map)# action forward
```

Step 6 Exit the VLAN access-map configuration mode using the command **exit** and configure an ERSPAN Type II source session using the command **monitor session** [*session-number* | **all**] **type erspan-source** [**shut**] .

Example:

```
switch(config-access-map)# exit
```

Example:

```
switch(config)# monitor session 1 type erspan-source
```

By default, the session is bidirectional. The optional keyword **shut** specifies a shut state for the selected session.

Step 7 Associate an ACL with the ERSPAN session using the command **filter access_group** *name* and display the ERSPAN ACL configuration using the command **show ip access-lists** *name* .

Example:

```
switch(config-erspan-src)# filter access_group erspan_filter
```

Example:

```
switch(config-acl)# show ip access-lists erpsan-acl
```

You can create an ACL using the standard ACL configuration process. For more information, see *Cisco Nexus 9000 Series NX-OS Security Configuration Guide.*

Step 8 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config-acl)# copy running-config startup-config
```

This is an optional step.

This example shows how to configure an ERSPAN ACL:

```
switch# configure terminal
switch(config)# ip access-list match_10_pkts
switch(config-acl)# permit ip 10.0.0.0/24 any
switch(config-acl)# exit
switch(config)# ip access-list match_172_pkts
switch(config-acl)# permit ip 172.16.0.0/24 any
switch(config-acl)# exit
```

In the case of different ERSPAN destinations where the interesting traffic is chosen based on the defined ACL filters, the last configured session would always have the higher priority.

For example, if Monitor Session 1 is configured; then Monitor Session 2 is configured; then ERSPAN traffic filter works as intended. But, if the user goes back to Monitor Session 1 and re-applies one of the existing configuration line (no new changes in the config); then the spanned traffic switches back to Monitor Session 1.

What to do next

To display the ERSPAN ACL configuration, execute the appropriate show commands from the following table.

Command	Purpose
show ip access-lists *name* Example: `switch(config-acl)# show ip access-lists erpsan-acl`	Displays the ERSPAN ACL configuration.
show vlan access-map *name* Example: `switch(config-acl)# show vlan access-map erspan_filter`	Displays information about VLAN access maps.

Command	Purpose
show monitor session {**all** | *session-number* | **range** *session-range*} [**brief**] Example: `switch(config-acl)# show monitor session 1`	Displays the ERSPAN session configuration.

Configure UDF-Based ERSPAN

Configure the device to match on user-defined fields (UDFs) of the outer or inner packet fields and send the matching packets to the ERSPAN destination. This helps analyze and isolate packet drops in the network.

You can configure the device to match on user-defined fields (UDFs) of the outer or inner packet fields (header or payload) and to send the matching packets to the ERSPAN destination. Doing so can help you to analyze and isolate packet drops in the network.

Before you begin

Make sure that the appropriate TCAM region (racl, ifacl, or vacl) has been configured using the **hardware access-list tcam region** command to provide enough free space to enable UDF-based ERSPAN. For information, see the "Configuring ACL TCAM Region Sizes" section in the Cisco Nexus 9000 Series NX-OS Security Configuration Guide.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Define the UDF using the command **udf** *udf-name offset-base offset length*

Example:

```
switch(config)# udf udf-x packet-start 12 1
 switch(config)# udf udf-y header outer l3 20 2
```

- *udf-name* —Specifies the name of the UDF. You can enter up to 16 alphanumeric characters for the name.
- *offset-base* —Specifies the UDF offset base as follows, where **header** is the packet header to consider for the offset: **packet-start** | **header** { **outer** | **inner** { **l3** | **l4** }} .
- *offset* —Specifies the number of bytes offset from the offset base. To match the first byte from the offset base (Layer 3/Layer 4 header), configure the offset as 0.
- *length* —Specifies the number of bytes from the offset. Only 1 or 2 bytes are supported. To match additional bytes, you must define multiple UDFs.

You can define multiple UDFs, but Cisco recommends defining only required UDFs.

Step 3 Attach the UDFs to one of the following TCAM regions using the command **hardware access-list tcam region** { **racl | ifacl | vacl** } **qualify udf** *udf-names*

Example:

```
switch(config)# hardware access-list tcam region
      racl qualify udf udf-x udf-y
```

- racl—Applies to Layer 3 ports.—Applies to layer 2 and Layer 3 ports.
- ifacl—Applies to Layer 2 ports.
- vacl—Applies to source VLANs.

You can attach up to 8 UDFs to a TCAM region.



Note When the UDF qualifier is added, the TCAM region goes from single wide to double wide. Make sure enough free space is available; otherwise, this command will be rejected. If necessary, you can reduce the TCAM space from unused regions and then re-enter this command. For more information, see the "Configuring ACL TCAM Region Sizes" section in the Cisco Nexus 9000 Series NX-OS Security Configuration Guide.



Note The **no** form of this command detaches the UDFs from the TCAM region and returns the region to single wide.

Step 4 Save the change persistently through reboots and restart by copying the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

Step 5 **reload**

Example:

```
switch(config)# reload
```

Reloads the device.



Note Your UDF configuration is effective only after you enter **copy running-config startup-config** + **reload** .

Step 6 Create an IPv4 access control list (ACL) and enters IP access list configuration mode using the command **ip access-list** *erspan-acl*

Example:

```
switch(config)# ip access-list erspan-acl-udf-only
 switch(config-acl)#
```

Step 7 Configure the ACL to match only on UDFs (example 1) or to match on UDFs along with the current access control entries (ACEs) for the outer packet fields (example 2). Enter one of the following commands to con:

- **permit udf** *udf-name value mask*

- **permit ip** *source destination* **udf** *udf-name value mask*

Example:

```
switch(config-acl)# permit udf udf-x 0x40 0xF0 udf-y 0x1001 0xF00F
```

Example:

```
switch(config-acl)# permit ip 10.0.0.0/24 any udf udf-x 0x02 0x0F udf-y 0x1001 0xF00F
```

A single ACL can have ACEs with and without UDFs together. Each ACE can have different UDF fields to match, or all ACEs can match for the same list of UDFs.

Step 8 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

This example shows how to configure UDF-based ERSPAN to match on the inner TCP flags of an encapsulated IP-in-IP packet using the following match criteria:

- Outer source IP address: 10.0.0.2
- Inner TCP flags: Urgent TCP flag is set
- Bytes: Eth Hdr (14) + Outer IP (20) + Inner IP (20) + Inner TCP (20, but TCP flags at 13th byte)
- Offset from packet-start: 14 + 20 + 20 + 13 = 67
- UDF match value: 0x20
- UDF mask: 0xFF

```
udf udf_tcpflags packet-start 67 1
hardware access-list tcam region racl qualify udf udf_tcpflags
copy running-config startup-config
reload
ip access-list acl-udf
permit ip 10.0.0.2/32 any udf udf_tcpflags 0x20 0xff
monitor session 1 type erspan-source
source interface Ethernet 1/1
filter access-group acl-udf
```

This example shows how to configure UDF-based ERSPAN to match regular IP packets with a packet signature (DEADBEEF) at 6 bytes after a Layer 4 header start using the following match criteria:

- Outer source IP address: 10.0.0.2
- Inner TCP flags: Urgent TCP flag is set
- Bytes: Eth Hdr (14) + IP (20) + TCP (20) + Payload: 112233445566DEADBEEF7788
- Offset from Layer 4 header start: 20 + 6 = 26
- UDF match value: 0xDEADBEEF (split into two-byte chunks and two UDFs)
- UDF mask: 0xFFFFFFFF

```
udf udf_pktsig_msb header outer l3 26 2
udf udf_pktsig_lsb header outer l3 28 2
hardware access-list tcam region racl qualify udf udf_pktsig_msb udf_pktsig_lsb
copy running-config startup-config
reload
ip access-list acl-udf-pktsig
permit udf udf_pktsig_msb 0xDEAD 0xFFFF udf udf_pktsig_lsb 0xBEEF 0xFFFF
monitor session 1 type erspan-source
source interface Ethernet 1/1
filter access-group acl-udf-pktsig
```

Configure ERSPAN Truncation

You can configure truncation for local and ERSPAN source sessions only.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Enter monitor configuration mode for the specified ERSPAN session using the command **monitor session** *session-number* **type erspan-source**

Example:

```
switch(config)# monitor session 10 type erspan-source
                        switch(config-erspan-src)#
```

Step 3 Configure the source interface using the command **source interface** *type slot* / *port* [**rx** | **tx** | **both**]

Example:

```
switch(config-erspan-src)# source interface ethernet 1/5 both
```

Step 4 Configure the MTU size for truncation using the command **mtu** *size*

Example:

```
switch(config-erspan-src)# mtu 512
```

Example:

```
switch(config-erspan-src)# mtu ?

<512-1518> Enter the value of MTU truncation size for
                        ERSPAN packets (erspan header + truncated original packet)
```

Any ERSPAN packet that is larger than the configured MTU size is truncated to the configured size. The MTU ranges for ERSPAN packet truncation are:

- The MTU size range is 64 to 1518 bytes for Cisco Nexus 9300-FX Series switches.
- The MTU size is 343 bytes (excluding FCS) for Cisco Nexus 9808 and 9804 platform switches.

Step 5 Configure the Ethernet ERSPAN destination port using the command **destination interface** *type slot* / *port*

Example:

```
switch(config-erspan-src)# destination interface Ethernet 1/39
```

Step 6 Enable the ERSPAN session using the command **no shut**

Example:

```
switch(config-erspan-src)# no shut
```

By default, the session is created in the shut state.

Step 7 Display the ERSPAN configuration using the command **show monitor session** *session*

Example:

```
switch(config-erspan-src)# show monitor session 5
```

This is an optional step.

Step 8 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config-erspan-src)# copy running-config startup-config
```

This example shows how to configure ERSPAN truncation for use with MPLS stripping:

```
mpls strip
ip access-list mpls
  statistics per-entry
  20 permit ip any any redirect Ethernet1/5

interface Ethernet1/5
  switchport
  switchport mode trunk
  mtu 9216
  no shutdown

monitor session 1
  source interface Ethernet1/5 tx
  mtu 64
  destination interface Ethernet1/6
  no shut
monitor session 21 type erspan-source
  description "ERSPAN Session 21"
  header-type 3
  erspan-id 21
  vrf default
  destination ip 10.1.1.2
  source interface Ethernet1/5 tx
  mtu 64
  no shut
monitor session 22 type erspan-source
  description "ERSPAN Session 22"
  erspan-id 22
  vrf default
  destination ip 10.2.1.2
  source interface Ethernet1/5 tx
```

```
  mtu 750
  no shut
monitor session 23 type erspan-source
  description "ERSPAN Session 23"
  header-type 3
  marker-packet 1000
  erspan-id 23
  vrf default
  destination ip 10.3.1.2
  source interface Ethernet1/5 tx
  mtu 1000
  no shut
```

Configuration Examples

This section helps understand the following

- Enable ERSPAN marker packets
- Configure a unidirectional ERSPAN session
- Shut Down or Activate an ERSPAN session

Enable ERSPAN Marker Packet

This example shows how to enable the ERSPAN marker packet with an interval of 2 seconds:

```
switch# configure terminal
switch(config)# monitor erspan origin ip-address 172.28.15.250 global
switch(config)# monitor session 1 type erspan-source
switch(config-erspan-src)# header-type 3
switch(config-erspan-src)# erspan-id 1
switch(config-erspan-src)# ip ttl 16
switch(config-erspan-src)# ip dscp 5
switch(config-erspan-src)# vrf default
switch(config-erspan-src)# destination ip 10.1.1.2
switch(config-erspan-src)# source interface ethernet 1/15 both
switch(config-erspan-src)# marker-packet 100
switch(config-erspan-src)# no shut
switch(config-erspan-src)# show monitor session 1
session 1
---------------
type              : erspan-source
state             : up
granularity       : nanoseconds
erspan-id         : 1
vrf-name          : default
destination-ip    : 10.1.1.2
ip-ttl            : 16
ip-dscp           : 5
header-type       : 3
origin-ip         : 172.28.15.250 (global)
source intf       :
    rx            : Eth1/15
    tx            : Eth1/15
    both          : Eth1/15
    rx            :
marker-packet     : enabled
```

```
packet interval   : 100
packet sent       : 25
packet failed     : 0
egress-intf       :
```

Configure a Unidirectional ERSPAN Session

This example shows how to configure a unidirectional ERSPAN session :

```
switch# configure terminal
switch(config)# interface ethernet 14/30
switch(config-if)# no shut
switch(config-if)# exit
switch(config)# no monitor session 3
switch(config)# monitor session 3 type erspan-source
switch(config-erspan-src)# source interface ethernet 2/1-3 rx
switch(config-erspan-src)# erspan-id 1
switch(config-erspan-src)# ip ttl 16
switch(config-erspan-src)# ip dscp 5
switch(config-erspan-src)# vrf default
switch(config-erspan-src)# destination ip 10.1.1.2
switch(config-erspan-src)# no shut
switch(config-erspan-src)# exit
switch(config)# show monitor session 1
```

Shut Down or Activate an ERSPAN Session

You can shut down ERSPAN sessions to discontinue the copying of packets from sources to destinations. You can shut down one session in order to free hardware resources to enable another session. By default, ERSPAN sessions are created in the shut state.

You can enable ERSPAN sessions to activate the copying of packets from sources to destinations. To enable an ERSPAN session that is already enabled but operationally down, you must first shut it down and then enable it. You can shut down and enable the ERSPAN session states with either a global or monitor configuration mode command.

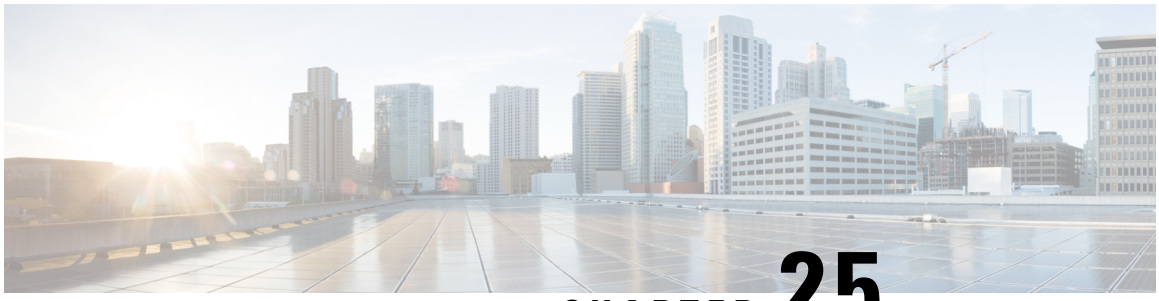
1. Enter the global configuration mode using the command **configure terminal**
2. Shut down the specified ERSPAN sessions using the command **monitor session** {*session-range* | **all**} **shut**. By default, sessions are created in the shut state.
3. Resume (enables) the specified ERSPAN sessions using the command **no monitor session** {*session-range* | **all**} **shut**. By default, sessions are created in the shut state.

Note If a monitor session is enabled but its operational status is down, then to enable the session, you must first specify the **monitor session shut** command followed by the **no monitor session shut** command.

4. Enter the monitor configuration mode for the ERSPAN source type using the command **monitor session** *session-number* **type erspan-source**. The new session configuration is added to the existing session configuration.
5. Shut down the ERSPAN session using the command **shut**. By default, the session is created in the shut state.
6. Enable the ERSPAN session using the command **no shut**. By default, the session is created in the shut state.

7. Exit the monitor configuration mode using the command **exit**.
8. Display the status of ERSPAN sessions using the command **show monitor session all**. This is an optional step.
9. Display the ERSPAN running configuration and the ERSPAN startup configuration using the commands **show running-config monitor** and **show startup-config monitor**. This is an optional step.

```
switch# configure terminal
switch(config)# monitor session 3 shut
switch(config)# no monitor session 3 shut
switch(config)# monitor session 3 type erspan-source
switch(config-erspan-src)# shut
switch(config-erspan-src)# no shut
switch(config-erspan-src)# exit
switch(config)# show monitor session all
switch(config)# show running-config monitor
switch(config)# show startup-config monitor
witch(config)# copy running-config startup-config
```



CHAPTER 25

Link Layer Discovery Protocol

Link Layer Discovery Protocol (LLDP) is a one-way protocol that transmits information about the capabilities and current status of a device and its interfaces. LLDP devices use the protocol to solicit information only from other LLDP devices.

- Overview, on page 425
- Guidelines and Limitations, on page 427
- Enable or Disable Globally, on page 429
- Enable or Disable on an Interface, on page 431
- Configure DCBXP Egress Queuing, on page 432
- Configure DCBXP Protocol Version, on page 433
- Enable or Disable Multi-Neighbor Support, on page 434
- Enable or Disable Support on Port-Channel Interfaces, on page 437
- Configure Optional Parameters, on page 439

Overview

The Cisco Discovery Protocol (CDP) and Link Layer Discovery Protocol (LLDP) are device discovery protocols that allow network management applications to automatically discover and learn about other devices connected to the network.

- CDP is a Cisco proprietary protocol for device discovery on the network.
- LLDP is a vendor-neutral protocol defined in IEEE 802.1ab, enabling discovery of non-Cisco devices.
- LLDP uses Type, Length, and Value (TLV) attributes to advertise configuration information, device capabilities, and device identity.

LLDP Advertised TLVs

LLDP advertises the following TLVs by default:

- DCBXP
- Management address
- Port description
- Port VLAN

- System capabilities
- System description
- System name

Data Center Bridging Exchange Protocol (DCBXP)

The Data Center Bridging Exchange Protocol (DCBXP) is an extension of LLDP. It is used to announce, exchange, and negotiate node parameters between peers. DCBXP parameters are packaged as DCBXP TLVs in the LLDP packet. If CEE is used, DCBXP will use an acknowledgment mechanism over LLDP. When the port comes up, DCBX TLVs are sent and any DCBX TLVs received are processed. By default, the DCBX protocol is set to auto-detect, and the latest protocol version supported by both the peers is used.

Features that need to exchange and negotiate parameters with peer nodes using DCBXP are as follows:

- Priority-based Flow Control (PFC)—PFC is an enhancement to the existing Pause mechanism in Ethernet. It enables Pause based on user priorities or classes of service. A physical link that is divided into eight virtual links with PFC provides the capability to use Pause on a single virtual link without affecting traffic on the other virtual links. Enabling Pause on a per-user-priority basis allows administrators to create lossless links for traffic requiring no-drop service while retaining packet-drop congestion management for IP traffic.
- Enhanced Transmission Selection (ETS)—ETS enables optimal bandwidth management of virtual links. ETS is also called priority grouping. It enables differentiated treatments within the same priority classes of PFC. ETS provides prioritized processing based on bandwidth allocation, low latency, or best effort, resulting in per-group traffic class allocation. For example, an Ethernet class of traffic may have a high-priority designation and a best effort within that same class. ETS allows differentiation between traffic of the same priority class, thus creating priority groups.
- Application Priority Configuration—Carries information about the priorities that are assigned to specific protocols.
- Priority to DSCP Mapping—The mapping of the DSCP and COS values configured in the QoS policy are sent in the Application Priority TLV.



Note For information on the quality of service (QoS) features, see the *Cisco Nexus 9000 Series NX-OS Quality of Service Configuration Guide*.

DCBXP is enabled by default, provided LLDP is enabled. When LLDP is enabled, DCBXP can be enabled or disabled using the [**no**] **lldp tlv-select dcbxp** command. DCBXP is disabled on ports where LLDP transmit or receive is disabled.

Beginning with Cisco NX-OS Release 10.2(3)F, an additional command is introduced: [**no**] **lldp tlv-select dcbxp egress-queuing**. While the [**no**] **lldp tlv-select dcbxp** command sends input queuing parameters in the ETS information that is exchanged with the peer, the [**no**] **lldp tlv-select dcbxp egress-queuing** command sends output queuing parameters in the ETS information. Hence, the bandwidths and priority information are extracted from the output queuing policy and exchanged with the peer.

At a time, you can configure either egress queuing or ingress queuing by running either **lldp tlv-select dcbxp egress-queuing** or **lldp tlv-select dcbxp** command as they overwrite each other.

The no forms of both the commands stop the DCBXP exchange on all interfaces.

To view which of the above two commands is enabled, run the **show lldp tlv-select** command.

When the default input queuing policy at system level is detached, the DCBXP exchange on all interfaces will stop sending ETS configuration and recommendation TLVs. However, the default output queuing policy at the system level cannot be detached.

Supported Features

- High Availability - The LLDP feature supports stateless and stateful restarts. After a reboot or supervisor switchover, the running configuration is applied.

  For more information on high availability, see the *Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide*.

- Virtualization Support - One instance of LLDP is supported.

Default Settings

This table lists the LLDP default settings.

Parameters	Default
Global LLDP	Disabled
LLDP on interfaces	Enabled, after LLDP is enabled globally
LLDP hold time (before discarding)	120 seconds
LLDP reinitialization delay	2 seconds
LLDP timer (packet update frequency)	30 seconds
LLDP TLVs	Enabled
LLDP receive	Enabled, after LLDP is enabled globally
LLDP transmit	Enabled, after LLDP is enabled globally
DCBXP	Enabled, provided LLDP is enabled
DCBXP version	Auto-detect

Guidelines and Limitations

- The LLDP feature must be enabled on the device before you can enable or disable the LLDP protocol on any interfaces.
- LLDP is supported only on physical interfaces.
- LLDP can discover up to one device per port.
- The Cisco Nexus 3232C and 3264Q switches do not support DCBXP.
- DCBXP incompatibility messages might appear when you change the network QoS policy on a switch, if a physical cable loopback connection is present on the device. The incompatibility exists for only a short time and then resolves on its own.

- PFC TLVs are sent when pause is enabled for at least one COS value in a network-qos policy and the priority-flow-control mode is set to auto at the interface level.
- DCBX TLVs are sent when ingress queuing is applied if **lldp tlv-select dcbxp** command is used and when output queuing is applied if **lldp tlv-select dcbxp egress-queuing** is used.
- The system name in LLDP can also display the domain name if it is configured. If the management VRF has a domain-name configured, then that is given higher priority over the default VRF domain-name. If no domain-name exists for the management VRF, then we check for default VRF domain-name and use that if its configured.

Release Specific Guidelines

Beginning with Release 10.1(1)

- Multiple LLDP neighbors per physical interface are supported on the following platforms:
  - N9K-C93180YC-FX3S
  - N9K-C93108TC-FX3P
  - N9K-C93180YC-FX3

Beginning with Cisco NX-OS Release 10.2(3)F

- The [**no**] **lldp tlv-select dcbxp egress-queuing** command is introduced to provide you the option to advertise egress queuing configuration in the switch. This feature is supported on Cisco Nexus 9200, 9300-EX, and 9300-FX platform switches.
- The Correctly Advertise LLDP Chassis-ID feature introduces a new global configuration command—**lldp chassis-id switch** —to advertise the switch chassis MAC address instead of the port MAC address, that is, all the ports will publish only the MAC address of the switch chassis. This feature is supported on all Cisco Nexus 9000 series platform switches.

Beginning with Cisco NX-OS Release 10.3(1)F

- LLDP (Multi Neighbor and Port- channel) is supported on the Cisco Nexus 9808 platform switches.
- PFC and ECN are supported on Cisco Nexus 9300-FX3 platform switches. On 9300-FX3 switches operating in FEX mode, DCBX is supported with ingress queuing.

Beginning with Cisco NX-OS Release 10.3(2)F

- New optional parameters such as 802.1 link aggregation, 802.1 vlan name, and 802.3 max frame size are enabled by default for the **lldp tlv-select** command. The VLAN name TLV only has the first 32 characters of the VLAN name, per the standards. Only the names of 25 active VLANs are displayed.

Beginning with Cisco NX-OS Release 10.4(1)F

- LLDP is supported on the following switches and line cards:
  - Cisco Nexus 9804 platform switch
  - Cisco Nexus X98900CD-A line card
  - Cisco Nexus X98900CD-A line card with Cisco Nexus 9808 and 9804 switches

Beginning with Cisco NX-OS Release 10.4(2)F

- LLDP is supported on Cisco Nexus 9232E-B1 platform switch.

Beginning with Cisco NX-OS Release 10.6(1)F

- LLDP is supported on Cisco N9336C-SE1 switch.

Platform Support for DCBXP

DCBXP is supported on the following platforms:

- Cisco Nexus 9200, 9300-EX, 9300-FX, 9300-FX2 and 9300-FX3 Series switches.
- Cisco Nexus , 9332PQ, , 9372PX, 9372PX-E, and 9396PX switches.
- Cisco Nexus 9504 and 9508 switches with X9432PQ, X9464PX, X9536PQ, X9564PX, X9636PQ line cards.

N9300 Series Smart Switch Guidelines

Beginning with NX-OS Release 10.6(2)F, LLDP is supported on N9324C-SE1U and N9348Y2C6D-SE1U Smart Switches.

Enable or Disable Globally

You can enable or disable LLDP globally on a device. You must enable LLDP globally to allow a device to send and receive LLDP packets.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Enable or disable LLDP on the device using the command [**no**] **feature lldp**

Example:

```
switch(config)# feature lldp
```

LLDP is disabled by default.

Note When this command is enabled, by default, the switch advertises the port MAC address per port.

Step 3 Enable this command to indicate that the switch chassis MAC address must be advertised for all the ports using the command [**no**] **lldp chassis-id switch**

Example:

```
switch(config)# lldp chassis-id switch
```

This is an optional step. Use the no form of this command to revert to advertising the port MAC address per port.



Note Use the **show vdc detail** command to view the switch chassis MAC address.

Step 4 Display the global LLDP configuration using the command **show running-config lldp**

Example:

```
switch(config)# show running-config lldp
```

This is an optional step. If LLDP is enabled, it shows "feature lldp." If LLDP is disabled, it shows an "Invalid command" error.

Step 5 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

This example shows how to enable LLDP on a device; disable LLDP on some interfaces; configure optional parameters such as hold time, delay time, and update frequency; and disable several LLDP TLVs:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# feature lldp
switch(config)# interface ethernet 7/9
switch(config-if)# no lldp transmit
switch(config-if)# no lldp receive
switch(config-if)# exit
switch(config)# interface ethernet 7/10
switch(config-if)# no lldp transmit
switch(config-if)# no lldp receive
switch(config-if)# exit
switch(config)# lldp holdtime 200
switch(config)# lldp reinit 5
switch(config)# lldp timer 50
switch(config)# no lldp tlv-select port-vlan
switch(config)# no lldp tlv-select system-name
```

What to do next

To display the LLDP configuration, perform one of the following tasks:

Command	Purpose
show running-config lldp	Displays the global LLDP configuration.
show lldp interface *interface slot/port*	Displays the LLDP interface configuration.
show lldp timers	Displays the LLDP hold time, delay time, and update frequency configuration.
show lldp tlv-select	Displays the LLDP TLV configuration.
show lldp neighbors {**detail** | **interface** *interface slot/port*}	Displays the LLDP neighbor device status. **Note** If the neighboring switch advertises switch MAC, then this show command displays the switch MAC and if it advertises the port MAC, the show command will display the port MAC.
show lldp traffic	Displays the LLDP counters, including the number of LLDP packets sent and received by the device, the number of discarded packets, and the number of unrecognized TLVs.
show lldp traffic interface *interface slot/port*	Displays the number of LLDP packets sent and received on the interface.
show qos dcbxp *interface slot/port*	Displays DCBXP information for a specific interface.

Use the **clear lldp counters** command to clear the LLDP statistics.

Enable or Disable on an Interface

Use this task to enable or disable LLDP on a specific interface after it has been globally enabled on the device.

After you globally enable LLDP, it is enabled on all supported interfaces by default. However, you can enable or disable LLDP on individual interfaces or selectively configure an interface to only send or only receive LLDP packets.

Note Cisco NX-OS commands for this feature may differ from Cisco IOS commands for a similar feature.

Before you begin

Make sure that you have globally enabled LLDP on the device.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Specify the interface on which you are enabling LLDP and enters the interface configuration mode using the command **interface** *interface slot / port*

Example:

```
switch(config)# interface ethernet 7/1
  switch(config-if)#
```

Step 3 Enable or disable the transmission of LLDP packets on an interface using the command [**no**] **lldp transmit**

Example:

```
switch(config-if)# lldp transmit
```

After you globally enable LLDP, it is enabled on all supported interfaces by default.

Step 4 Enable or disable the reception of LLDP packets on an interface using the command [**no**] **lldp receive**

Example:

```
switch(config-if)# lldp receive
```

After you globally enable LLDP, it is enabled on all supported interfaces by default.

Step 5 Display the LLDP configuration on the interface using the command **show lldp interface** *interface slot / port*

Example:

```
switch(config-if)# show lldp interface ethernet 7/1
```

This is an optional step.

Step 6 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure DCBXP Egress Queuing

Use the following procedure to configure DCBXP egress queuing.

Before you begin

- Make sure that you have globally enabled LLDP on the device (global configuration command **feature lldp**).

Note After you globally enable LLDP, it is enabled on all supported interfaces by default.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Enable DCBXP TLVs globally and starts sending input queuing parameters in the ETS information exchanged with the peer using the command **lldp tlv-select dcbxp**

Example:

```
switch(config)# lldp tlv-select dcbxp
                        switch(config)#
```

Step 3 Enable DCBXP TLVs globally and starts sending output queuing parameters in the ETS information using the command **lldp tlv-select dcbxp egress-queuing**

Example:

```
switch(config)# lldp tlv-select dcbxp egress-queuing
                        switch(config)#
```

This is an optional step.

Configure DCBXP Protocol Version

You can specify the protocol version in which the DCBX TLVs are sent.

- Nexus 9000 switches do not support auto-negotiating or hard coding DCBXP TLV using CIN version.
- If the peers are not running the same version, DCBX parameters may not converge for the link. Reset the link for the new protocol version to take effect.

Before you begin

Make sure that you have globally enabled LLDP on the device.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Enter interface configuration mode using the command **interface** *interface slot* / *port*

Example:

```
switch(config)# interface ethernet 1/25
                        switch(config-if)#
```

Step 3 Specify the protocol version mode sent using the command **lldp dcbx version** *cee* / *ieee* / *auto*

Example:

```
switch(config-if)#lldp dcbx version cee
```

- The *cee* variable sets the port to only send TLVs in Converged Enhanced Ethernet (CEE) protocol version.
- The *ieee* variable sets the port to only sent TLVs in IEEE 802.1Qaz protocol version.
- The *auto* variable sets the port to send TLVs in the latest protocol version supported by both the peers.

The default is set to *auto* .

Note Devices that do not support IEEE 802.1Qaz may not properly respond to auto-negotiation attempts and may require the interface to be manually configured for lldp dcbx version cee.

Enable or Disable Multi-Neighbor Support

Before you begin

Often times a network device sends multiple LLDP packets, out of which one is from the actual host. If a Cisco Nexus switch is communicating with the device but can only manage a single LLDP neighbor per interface, there is a good chance that becoming a neighbor with the actual required host will fail. To minimize this, Cisco Nexus switch interfaces can support multiple LLDP neighbors creating a better opportunity of becoming an LLDP neighbor with the correct device.

Support for multiple LLDP neighbors over the same interface requires LLDP multi-neighbor support to be configured globally.

Note You must disable DCBX globally before configuring LLDP multi-neighbor support. Failure to do so invokes an error message.

Consider the following before enabling LLDP multi-neighbor support on the interfaces:

- Make sure that you have globally enabled LLDP on the device (global configuration command **feature lldp**).

Note After you globally enable LLDP, it is enabled on all supported interfaces by default.

- A maximum of three (3) neighbors are supported on an interface.
- LLDP multi-neighbor is not supported on FEX interfaces.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
                        switch(config)#
```

Note If the output of the **show lldp tlv-select** command shows dcbxp, then perform step 2 and skip step 3. If the output shows dcbxp egress-queuing then skip step 2 and perform step 3.

This is required to avoid invoking an error message when you configure LLDP multi-neighbor support.

Step 2 Disable DCBXP TLVs globally using the command **no lldp tlv-select dcbxp**

Example:

```
switch(config)# no lldp tlv-select dcbxp
                        switch(config)#
```

Step 3 Disable DCBXP TLVs globally using the command **no lldp tlv-select dcbxp egress-queuing**

Example:

```
switch(config)# no lldp tlv-select dcbxp egress-queuing
                        switch(config)#
```

Step 4 Enable or disable LLDP multi-neighbor support for all interfaces globally using the command [**no**] **lldp multi-neighbor**

Example:

```
switch(config)# lldp multi-neighbor
                        switch(config)#
```

Step 5 Specify the interface on which you will enable LLDP and enter the interface configuration mode using the command **interface** *port* / *slot*

Example:

```
switch(config)# interface 1/1
                        switch(config-if)#
```

Step 6 Disable (or enable) the transmission of LLDP packets on the interface using the command [**no**] **lldp transmit**

Example:

```
switch(config-if)# lldp transmit
```

This is an optional step.



Note The transmission of LLDP packets on this interface was enabled using the global **feature lldp** command. This option is to disable the feature for this specific interface.

Step 7 Disable (or enable) the reception of LLDP packets on the interface using the command [**no**] **lldp receive**

Example:

```
switch(config-if)# lldp receive
```

This is an optional step.



Note The reception of LLDP packets on this interface was enabled using the global **feature lldp** command. This option is to disable the feature for this specific interface.

Step 8 Display the LLDP configuration on the interface using the command **show lldp interface** *port* / *slot*

Example:

```
switch(config-if)# show lldp interface 1/1
```

This is an optional step.

Step 9 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Enable or Disable Support on Port-Channel Interfaces

Use this procedure to enable or disable LLDP support on port-channel interfaces.

LLDP must be globally enabled before it can be configured on port-channel interfaces. Applying LLDP configuration commands to a port-channel does not affect its member interfaces. LLDP neighbors form between port-channels only when LLDP transmit and receive are configured on both sides. LLDP transmit and receive commands do not work on MCT, VPC, fex-fabric, FEX port-channels, or port-channel sub-interfaces.

- After you globally enable LLDP, it is enabled on all supported interfaces by default.
- If you enable the LLDP port-channel feature globally, the LLDP configuration is not applied to MCT, VPC, fex-fabric, FEX port-channels, or port-channel sub-interfaces. If the configuration is removed from the port-channels or the port type feature is disabled globally, you cannot use the **lldp port-channel** command to enable it on the newly supported port-channels. To enable LLDP port-channel on the port-channels in question, configure **lldp transmit** and **lldp receive** for each port-channel.

Before you begin

Consider the following before enabling LLDP support on port-channels:

- Make sure that you have globally enabled LLDP on the device (global configuration command **feature lldp**).
- Applying the **lldp transmit** and **lldp receive** configuration commands to a port-channel does not affect the configuration for the members of the port-channel.
- LLDP neighbors form between the port-channels only when LLDP transmit and receive is configured on both sides of the port-channel.
- The LLDP transmit and receive commands do not work on MCT, VPC, fex-fabric, FEX port-channels, and port-channel sub-interfaces.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
            switch(config)#
```



Note If the output of the **show lldp tlv-select** command shows dcbxp, then perform step 2 and skip step 3. If the output shows dcbxp egress-queuing then skip step 2 and perform step 3.

This is required before configuring LLDP on port-channels.

Step 2 Disable DCBXP TLVs globally using the command **no lldp tlv-select dcbxp**

Example:

```
switch(config)# no lldp tlv-select dcbxp
            switch(config)#
```

Step 3 Disable DCBXP TLVs globally using the command **no lldp tlv-select dcbxp egress-queuing**

Example:

```
switch(config)# no lldp tlv-select dcbxp egress-queuing
            switch(config)#
```

Step 4 Enable or disable LLDP transmit and receive for all port channels globally using the command [**no**] **lldp port-channel**

Example:

```
switch(config)# lldp port-channel
            switch(config)#
```

Step 5 Specify the interface port-channel on which you will enable LLDP and enter the interface configuration mode using the command **interface port-channel** [*port-channel-number* | *port-channel-range*]

Example:

```
switch(config)# interface port-channel 3
            switch(config-if)#
```

Example:

Enter a range of port-channel numbers if you are configuring LLDP over more than one port-channel:

```
switch(config)# interface port-channel 1-3
            switch(config-if-range)#
```

Specifies the interface port-channel range on which you are enabling LLDP and enters the interface range configuration mode.

Step 6 Disable (or enable) the transmission of LLDP packets on the port-channel or range of port-channels using the command [**no**] **lldp transmit**

Example:

```
switch(config-if)# lldp transmit
```

This is an optional step.



Note The transmission of LLDP packets on this port-channel was enabled using the global **lldp port-channel** command in step 3. This option is to disable the feature for this specific port-channel.

Step 7 Disable (or enable) the reception of LLDP packets on the port-channel or range of port-channels using the command [**no**] **lldp receive**

Example:

```
switch(config-if)# lldp receive
```

This is an optional step.



Note The reception of LLDP packets on this port-channel was enabled using the global **lldp port-channel** command in step 3. This option is to disable the feature for this specific port-channel.

Step 8 Display the LLDP configuration on the port-channel using the command **show lldp interface port-channel** l *port-channel-number*

Example:

```
switch(config-if)# show lldp interface port-channel 3
```

This is an optional step.

Step 9 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure Optional Parameters

You can configure the frequency of LLDP updates, the amount of time for a receiving device to hold the information before discarding it, and the initialization delay time. You can also select the TLVs to include in LLDP packets.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Specify the amount of time in seconds that a receiving device should hold the information that is sent by your device before discarding it using the command [**no**] **lldp holdtime** *seconds*

Example:

```
switch(config)# lldp holdtime 200
```

This is an optional step. The range is 10 to 255 seconds; the default is 120 seconds.

Step 3 Specify the delay time in seconds for LLDP to initialize on any interface using the command [**no**] **lldp reinit** *seconds*

Example:

```
switch(config)# lldp reinit 5
```

This is an optional step. The range is 1 to 10 seconds; the default is 2 seconds.

Step 4 Specify the transmission frequency of LLDP updates in seconds using the command [**no**] **lldp timer** *seconds*

Example:

```
switch(config)# lldp timer 50
```

This is an optional step. The range is 5 to 254 seconds; the default is 30 seconds.

Step 5 Display the LLDP hold time, delay time, and update frequency configuration using the command **show lldp timers**

Example:

```
switch(config)# show lldp timers
```

This is an optional step.

Step 6 Specify the TLVs to send and receive in LLDP packets using the command [**no**] **lldp tlv-select** *tlv*

Example:

```
switch(config)# lldp tlv-select system-name

switch(config)# lldp tlv-select port link-aggregation
```

This is an optional step. The available TLVs are management-address, port-description, port-vlan, system-capabilities, system-description, port vlan-name, port link-aggregation, port max-framesize, and system-name. All available TLVs are enabled by default. Use the no form of the command to disable the TLV.

Step 7 Display the LLDP TLV configuration using the command **show lldp tlv-select**

Example:

```
switch(config)# show lldp tlv-select
```

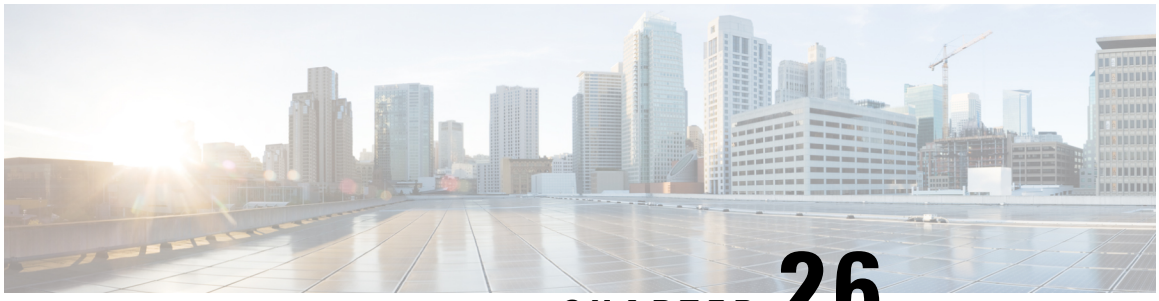
This is an optional step.

Step 8 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.



CHAPTER 26

NetFlow

A NetFlow system is a network traffic monitoring and reporting solution that

- identifies, classifies, and records packet flows traversing Nexus 9000 series switches,
- provides detailed flow-based statistics for IP and Layer 2 packet data, and
- supports advanced features for accounting, network performance, and security analysis without direct modification to network traffic or devices.

NetFlow does not require any change to the packets or to the networking device. Cisco NX-OS supports the flexible NetFlow feature that enables enhanced network anomalies and security detection. Flexible NetFlow allows you to define an optimal flow record for a particular application by selecting the keys from a large collection of predefined fields. Cisco NX-OS can gather NetFlow statistics and analyze all packets on the interface or subinterface.

You can export the data that NetFlow gathers for your flow by using a flow exporter and export this data to a remote NetFlow Collector, such as Cisco Stealthwatch. Cisco NX-OS exports a flow as part of a NetFlow export User Datagram Protocol (UDP) under the following circumstances:

- Flows are exported periodically as per the flow timeout value, which defaults to 10 seconds if not configured.
- You have forced the flow to export.

A flow is a unidirectional stream of packets that arrives on a source interface (or VLAN) and has the same values for the keys. A key is an identified value for a field within the packet. You can create a flow using a flow record to define the unique keys for your flow. All key values must match for the packet to count in a given flow. A flow might gather other fields of interest, depending on the export record version that you configure. The flow record determines the size of the data to be collected for a flow. Flows are stored in the NetFlow cache. The flow monitor combines the flow record and flow exporter with the NetFlow cache information.

- NetFlow Fundamentals, on page 442
- Ingress NetFlow, on page 445
- Egress NetFlow, on page 453
- Sampled Netflow, on page 455
- General Procedures to Enable NetFlow, on page 457
- Additional Procedures for Ingress NetFlow, on page 464
- Additional Procedures for Egress NetFlow, on page 468
- Drop Reports, on page 469

• Dropped flow notification, on page 472

NetFlow Fundamentals

Key Components

Table 32: Key Components of Netflow

Type	Definition
Flow Records	A flow record defines the keys that NetFlow uses to identify packets and other fields of interest that NetFlow gathers for the flow. You can define a flow record with any combination of keys and fields of interest. Cisco NX-OS supports a rich set of keys. A flow record also defines the types of counters gathered per flow. You can configure 32- or 64-bit packet or byte counters. The key fields are specified with the **match** keyword. The fields of interest and counters are specified under the **collect** keyword. Cisco NX-OS enables the following match fields as the defaults when you create a flow record: • match interface input • match flow direction
Flow Monitors	A flow monitor references the flow record and flow exporter. You apply a flow monitor to an interface.

Type	Definition
Flow Exporters	A flow exporter contains network layer and transport layer details for the NetFlow export packet. You can configure the following information in a flow exporter: • Export destination IP address • Source interface • UDP port number (where the NetFlow Collector is listening for NetFlow packets)—The default value is 9995. **Note** NetFlow export packets use the IP address that is assigned to the source interface. If the source interface does not have an IP address assigned to it, the flow exporter drops flows that were meant to be exported. The Netflow Exporter source interface and destination IP must use the same VRF. Cisco NX-OS exports data to the NetFlow Collector whenever a timeout occurs. You can configure a flush cache timeout (using the **flow timeout** command) to flush the cache and force a flow export.
Layer 2 NetFlow Keys	You can define Layer 2 keys in flexible NetFlow records that you can use to capture flows in Layer 2 interfaces. The Layer 2 keys are as follows: • Source and destination MAC addresses • Source VLAN ID • EtherType from the Ethernet frame You can apply Layer 2 NetFlow to the following interfaces for the ingress direction: • Switch ports in access mode • Switch ports in trunk mode • Layer 2 port channels **Note** You cannot apply Layer 2 NetFlow to VLANs, egress interfaces, or Layer 3 interfaces such as VLAN interfaces.

Type	Definition
NetFlow Output Interface	The NetFlow output interface on Cisco Nexus switches have the following features: • NetFlow in the **show flow cache** command displays `output_if_id` and exports output interface to the collector. • The NetFlow output interface for Cisco Nexus platform switches supports both IPv4 and IPv6 traffic flows. However, the NetFlow output interface for Cisco Nexus 9500 platform switches is supported only for IPv4 traffic flows and is not supported for IPv6 traffic flows. • The **show flow cache** command displays `output_if_id` as `0x0`. However, beginning with Cisco NX-OS Release 10.3(3)F, the **show flow cache** command displays `output_if_id` as `0x0` only if the destination IP address cannot be resolved or is not installed in the routing table or the packet received is control packet. • NetFlow supports exporting output interface to the collector for IPv4/IPv6 incoming traffic flows, which have Next-Hop as destination interface. The NetFlow export format for `InputInt` and `OutputInt` support the full 32-bit SNMP `ifIndex` values at the NetFlow Collector. • The NetFlow output interface is not supported for tunnel traffic flows such as MPLS, VXLAN, and GRE. • For more information on examples for NetFlow output interface, see the Display example for Netflow.

Note Cisco NX-OS supports the Version 9 export format. This format supports a more efficient network utilization than the older Version 5 export format and supports IPv6 and Layer 2 fields. In addition, the Version 9 export format supports the full 32-bit SNMP ifIndex values at the NetFlow Collector.

Note Cisco NX-OS supports stateful restarts for NetFlow. After a reboot, Cisco NX-OS applies the running configuration. The flow cache is not preserved across restarts, and packets that come to the software during restarts cannot be processed.

Architecture

Cisco Nexus 9000 Series switches separate NetFlow processing into two layers:

- The first layer supports per-packet visibility for line-rate traffic. Packets do not need to be sampled and statistically analyzed. Instead, the packets can be processed and aggregated at line rate.
- The second layer enables the gathering of flows at scale. It can maintain hundreds of thousands of flows without losing any flows and periodically exports them to an external collector.

Types

From Cisco NX-OS Release 10.4(1)F, NetFlow consists of two types configurations:

Table 33: Types of Netflow

Type	Definition
Ingress Netflow	Ingress NetFlow identifies packet flows for ingress IP packets and provides statistics based on the packet flows. Ingress NetFlow uses flows to provide statistics for accounting, network monitoring, and network planning.
Egress Netflow	Egress NetFlow identifies packet flows for outgoing IP packets and Layer 2 flows and provides statistics based on these packet flows. On Cisco Nexus Cloudscale 9000 switches, egress NetFlow uses ingress pipeline and ingress TCAM carving egr-netflow to record flow information egressing out of the switch. All parameters and restrictions of NetFlow such as flow records, flow exporters, and flow monitors are applicable to egress NetFlow.

Ingress NetFlow

Guidelines and Limitations

Note For verified NetFlow scalability numbers, see the Cisco Nexus 9000 Series NX-OS Verified Scalability Guide.

General Limitations

- NetFlow is not supported on tunnel interfaces.
- NetFlow is not supported for CPU-transmitted packets.
- Flow collection is not performed for ARP traffic.
- Enable NetFlow only on platforms that support this feature.

- The **match ip tos** command is present in flow record configuration options, but the functionality is not supported.
- NetFlow is not supported on loopback and switch management interfaces.
- Netflow and Flow Telemetry are not supported in N9K-C9364C-H1 platform SFP+ ports, Ethernet1/65, and Ethernet1/66.
- The Cisco Nexus 3232C and 3264Q switches do not support NetFlow.
- NetFlow for MPLS/VXLAN datapath is not supported
- TCAM carving configuration of the ing-netflow region can be performed on FX line cards. EX line cards have a default ing-netflow region TCAM carving of 1024 and cannot be configured otherwise. For ports on the EX and FX line cards, the suggested maximum for the ing-netflow region is 1024.
- Flows are identified with Ingress(0) field in CFLOW packet.

ICMP and Non-TCP/UDP Flows

- On Nexus 9000 Switches, NetFlow collects ICMP flow information and sends it to the collector. ICMP type and code are inherently included in the packets. ICMP packets in NetFlow or flow export records do not use traditional source and destination ports like TCP or UDP. Instead, the exporter often encodes the ICMP type and code into fields like SPORT (source port) and DPORT (destination port). For example:
  - ICMP Echo Request: SPORT=2048, DPORT=0
  - ICMP Echo Reply: SPORT=0, DPORT=0
  - ICMP Time Exceeded: SPORT=2816, DPORT=0

Configuration and Operational Guidelines

- Flow cache can be cleared per flow type, such as Layer 2, IPv4, and IPv6. It cannot be cleared per flow monitor.
- You must configure a source interface for the NetFlow Data Export (NDE). If you do not configure a source interface, the flow exporter drops flows that were meant to be exported.

Layer 2 and Layer 3 Interface Limitations

- Layer 2 switched flow monitors are applied only to Layer 2 interfaces.
- IP and IPv6 flow monitors can be applied to VLANs, SVIs, Layer 3 routed interfaces, or subinterfaces.
- If you change a Layer 2 interface to a Layer 3 interface, or vice versa, the software removes the Layer 2 NetFlow configuration from the interface.
- The same flow monitor cannot be shared with a VLAN and Layer 3 interfaces (for example, physical Layer 3 interface, SVI interface, or Layer 3 subinterface).
- VLAN and Layer 3 interfaces use different ACLs and must be treated as two different profiles.
- If you try to modify a record that is programmed in hardware during a rollback, the rollback fails.

Release Specific Guidelines

- *Cisco NX-OS Release 9.2(1)*

- NetFlow for FEX Layer 3 ports is supported on Cisco Nexus 9300-FX platform switches.

- *Cisco NX-OS Release 9.2(2)*
  - NetFlow is supported on Cisco Nexus 9500 platform switches with Cisco Nexus 9700-EX line cards and FM-E modules.

- *Cisco NX-OS Release 9.3(1)*
  - Commands **record netflow ipv4 original-input**, **record netflow ipv4 original-output**, and **record netflow layer2-switched input** are not supported.
  - The NetFlow **output_if_id** is supported on 9300-FX switches.

- *Cisco NX-OS Release 9.3(3)*
  - The following Non-Disruptive In-Service Software Upgrade (ND ISSU) limitations about NetFlow apply for all Cisco Nexus 9000 Series switches:
    - While performing an ND ISSU, a two-minute export loss is expected.
    - During an ND ISSU, an exporter with a management interface source port is not supported. Export loss is expected until the management interface comes up.
  - Ingress NetFlow is supported on Cisco Nexus 9300-GX platform switch.
  - The NetFlow **output_if_id** is supported on 9300-FX3 switches.

- *Cisco NX-OS Release 9.3(4)*
  - The RTP monitoring feature enables a monitor of RTP flows on all interfaces of a switch and reports them in the **show flow rtp detail** command output. An RTP flow is any UDP flow with a source port within the range of 16384-32767. If a NetFlow monitor is attached to a switch interface with RTP monitoring enabled, then all the traffic/flows (including the RTP flows) on that interface are reported in the output of the **show flow cache** command. The RTP flows will no longer be shown in the output of the **show flow rtp detail** command. When the attached monitor is removed, the RTP flows are reported again in the **show flow rtp detail** command output.

    This limitation impacts the following switches:
    - Cisco Nexus 9336C-FX2
    - Cisco Nexus 93240YC-FX2
    - Cisco Nexus 9348GC-FXP
    - Cisco Nexus 93180YC-FX
    - Cisco Nexus 93108TC-FX
    - Cisco Nexus 9316D-GX
    - Cisco Nexus 93600CD-GX
    - Cisco Nexus 9504, 9508, and 9516 with the 9736C-FX line card

- *Cisco NX-OS Release 10.1(2)*

- Netflow is supported on N9K-X9716D-GX line card.
-
  - You can attach either Layer 3 flow monitor or Layer 2 flow monitor to Layer 2 interface, not both.
  - If a flow monitor is already attached to Layer 3 interface, then the same flow monitor cannot be attached to Layer 2 interface.
  - The **mac-packet-classify** command is not supported, when Layer 3 flow monitor is applied on Layer 2 interface.
- Ingress NetFlow is supported on Cisco Nexus 9500 platform switches with EX, FX, and GX mixed chassis. You can use SPAN simultaneously with NetFlow on the Cisco Nexus 9500 platform switches with EX, FX, and GX mixed chassis. Cisco Nexus 9500-GX platform switches does not support SPAN with sFlow feature mix.

- *Cisco NX-OS Release 10.3(3)F*
  - Netflow Output_if_id is supported on Cisco Nexus 9300-FX2, 9300-GX, and 9300-GX2 switches and 9500-GX Line Cards.
    - For unicast flows, if the output_if_id is shown as 0x0, it means that the destination IP is not resolved or not available in the routing table or the ARP table.
    - The output_if_id is shown as 0x0 for received packets destined to the control-plane. Examples of such traffic are routing protocols or ICMP packets.
    - For IPv6 flows on Cisco Nexus 9500 Line Cards, output_if_id is shown as Unsupported due to hardware limitations.
    - The output_if_id for data plane multicast traffic will be shown as 0x0.
    - NetFlow does not export a flow record when the ingress and output interface IDs are the same. This behavior can reduce the number of exported records on a Layer 2 switch that is used for TAP aggregation.

      Use the show flow cache command to compare the if_id and output_if_id fields. If the fields are the same and these flows must be exported, disable MAC address learning globally by using the mac-learn disable command, and then clear dynamically learned MAC addresses by using the clear mac address-table dynamic command.

      For more information, see “Disabling MAC Address Learning on System” in the **Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide**.

- *Cisco NX-OS Release 10.4(1)F*
  - Ingress NetFlow is supported on N9K-C9332D-H2R switch. Drop flows is not supported by default. To enable drop flows, configure the **hardware flow-table collect-drop-reason** command.

- *Cisco NX-OS Release 10.4(2)F*
  - The Ingress NetFlow for MPLS packet feature is supported on Cisco Nexus 9300-FX, 9300-FX2, 9300-FX3, 9300-GX, 9300-GX2, and 9500-FX/GX switches. This feature allows the Nexus switches to capture IPv4 or IPv6 flow information from MPLS packets and send the flow information to the collector. NetFlow captures information for packets with up to 8 MPLS stacked labels and supports capturing of MPLS packets such as IPoMPLS and EoMPLS packets and MPLS-VPN packets with

two labels. The show flow cache shows output interface on MPLS-VPN encapsulation node as 0x0. The NetFlow output interface supports MPLS-VPN on transit or P nodes. This feature has the following limitations:

- Due to the hardware ASIC limitations, NetFlow cannot capture MPLS related parameters such as MPLS label.
- On transit or P nodes, MPLS VPNs with overlapping IP addresses with the same Layer 2 source and destination ports are reported as a single flow in the MPLS-VPN network.

- Ingress NetFlow is supported on N9K-C93400LD-H1 switch. Drop flows is not supported by default. To enable drop flows, configure the **hardware flow-table collect-drop-reason** command.

- *Cisco NX-OS Release 10.6(2)F*
  - Nexus 9300-SG2 switches support dropped flow notification feature.
  - Ingress NetFlow is supported on Nexus N9336C-SE1, N9396T12C-SE1, N9396Y12C-SE1, N9348Y12C-SE1, N9324C-SE1U, and N9348Y2C6D-SE1U switches.
  - Ingress NetFlow does not support hardware rate-limiter.
  - The show flow cache stores up to 1,000,000 flows. However, due to ASIC limitations, the byte count displayed in the output of this command is always zero (0) and the collector also receives a byte count of zero (0).
  - Output interfaces are not supported and are shown as 0x0 in the output of the show flow cache command.
  - The switches do not capture the control packets such as OSPF/BGP and self-IP packets such as BFD.
  - A single flow monitor cannot be used on Layer 3 and SVI interfaces simultaneously.
  - Up to 5 interface types are supported, and you can configure up to 14 profiles per interface type.
  - A Layer 3 flow monitor configured on Layer 3 interface cannot be applied to Layer 2 interfaces. Similarly, a Layer 2 flow monitor configured on Layer 2 interface cannot be applied to Layer 3 interfaces.
  - If a packet is denied by a Router Access Control List (RACL), the denied packet is still visible in the show flow cache records.
  - Ingress NetFlow allows 14 unique monitors to be active per acl key profiles. For ingress NetFlow there are 9 different acl key profiles pre-configured including all key types such as IPv4, IPv6 and MAC.

VXLAN Environment Limitations

- NetFlow for VXLAN traffic is supported including OIF (output interface). The feature nv overlay command must be enabled on the VXLAN transit node to capture inner flow details. If not, NetFlow captures only the outer header information.
- OIF on VXLAN Encapsulation node is shown as 0x0.
- NetFlow is supported on SVI and non-uplink L3 Interfaces of a VXLAN VTEP. This does not include the L3VNI SVI.

- NetFlow is not supported on uplink interfaces on a VXLAN VTEP.
- NetFlow on Multisite Border Gateways is not supported.
- A NetFlow Collector that is reachable over the VXLAN fabric is supported.

Platform Limitations

- Ingress NetFlow is not supported on Cisco Nexus 92348GC-X platform switch.
- Record match that is based on IP ToS, is not supported for IPv6 flow monitors. The ToS value is collected on the collector as 0x0 irrespective of the value the traffic holds. This limitation is applicable for the following platform switch families:
  - Cisco Nexus 9300-FX
  - Cisco Nexus 9300-FX2
  - Cisco Nexus 9300-FX3
  - Cisco Nexus 9300-GX
  - Cisco Nexus 9500 with EX and FX line cards
- *Cisco Nexus 9300-FX Platform Switches*
  - NetFlow VLANs are supported for switched traffic only, and NetFlow SVIs are supported for routed traffic only.
  - If you add a member to a port channel that is already configured for Layer 2 NetFlow, its NetFlow configuration is removed and the Layer 2 configuration is added.
- Cisco Nexus 9300-FX2 platform switches support sFlow and SPAN coexistence.
- On Cisco Nexus 9300-FX platform switches, and Cisco Nexus 9500 platform switches with EX/FX modules, SPAN, and sFlow cannot both be enabled simultaneously. If one is active, the other cannot be enabled. However, on the Cisco Nexus 9300-FX/FX2 and the Cisco Nexus 9500 platform switches with EX modules, both NetFlow and SPAN can be enabled simultaneously, providing a viable alternative to using sFlow and SPAN.
- Configuring an EX port as a trunk when FX ports are trunks with NetFlow configurations already applied, does not remove the unsupported EX NetFlow configuration from the FX port trunks. For example, if you apply more than two different IPv4 flow monitors to FX port trunks and if EX ports are added to the same trunks, the configuration on the trunks beyond the two monitors is not automatically removed, since it's only an EX port limitation. Since this configuration will not report flows beyond two monitors for EX trunk ports, we recommend that you use only two monitors per protocol (v4/v6/CE) on modular switches that could potentially have both EX and FX ports in the same trunk. This applies to all Cisco Nexus 9500 platform switches with EX and FX line cards.
- NetFlow is supported on Cisco Nexus 9500 platform switches with EX and FX mixed chassis. You can use SPAN simultaneously with NetFlow on the Cisco Nexus 9500 platform switches with EX and FX mixed chassis.
- Cisco Nexus 9500 platform switches with FM-E, FM-E2, and FM-E3 modules and Cisco Nexus 9300-FX/FX3 switches support the NetFlow output interface feature. However, output interface is not supported 9500-EX platform switches.

Prerequisites

- Make sure that you understand the resources required on your device because NetFlow consumes memory and CPU resources.

Configure Ingress NetFlow

Procedure

Step 1 Enable the NetFlow feature.

Step 2 Define a flow record by specifying keys and fields to the flow.

Step 3 Define an optional flow exporter by specifying the export format, protocol, destination, and other parameters.

Step 4 Define a flow monitor based on the flow record and flow exporter.

Step 5 Apply the flow monitor to a source interface , subinterface, or VLAN interface .

Configuration Example for Ingress NetFlow

This example shows how to configure an Ingress NetFlow exporter configuration for IPv4:

```
feature netflow
flow exporter ee
 destination 171.70.242.48 use-vrf management
 source mgmt0
 version 9
  template data timeout 20
flow record rr
 match ipv4 source address
 match ipv4 destination address
 collect counter bytes
 collect counter packets
flow monitor foo
 record rr
 exporter ee
interface Ethernet2/45
 ip flow monitor foo input
 ip address 10.20.1.1/24
 no shutdown
```

Ingress NetFlow Verification Commands

An Ingress NetFlow verification command is a diagnostic command that

- displays the status and configuration of Ingress NetFlow features on a device,
- provides detailed information on flows, interfaces, exporters, and records, and
- helps administrators troubleshoot and validate NetFlow operations.

To display the Ingress NetFlow configuration, perform one of the following tasks:

Command	Purpose
show flow cache [**ipv4** | **ipv6** | **ce**]	Displays information about Ingress NetFlow IP flows. **Note** This command can appear to be not valid on the EOR switches and no flows can be seen. To view this command on the EOR switches, attach to the module using the **attach mod** *x* command or check this command using the **slot** *x* **quoted "show flow cache"** command where *x* is the module number of the ingress NetFlow.
show flow exporter [*name*]	Displays information about Ingress NetFlow flow exporters and statistics. You can enter up to 63 alphanumeric characters for the flow exporter name.
show flow interface [*interface-type slot/port*]	Displays information about Ingress NetFlow interfaces.
show flow record [*name*]	Displays information about Ingress NetFlow flow records. You can enter up to 63 alphanumeric characters for the flow record name.
show flow record netflow layer2-switched input	Displays information about the Layer 2 Ingress NetFlow configuration.
show running-config netflow	Displays the NetFlow configuration that is currently on your device.

Use the **show flow exporter** command to display NetFlow statistics. Use the **clear flow exporter** command to clear NetFlow flow exporter statistics.

Display Example for Ingress NetFlow

The output of the **show flow cache** command for IPv4 displays:

```
show flow cache
IPV4 Entries
SIP          DIP         BD ID  S-Port  D-Port Protocol Byte Count  Packet Count  TCP FLAGS
  TOS  if_id      output_if_id  flowStart flowEnd
10.10.30.4   30.33.1.2   1480   30000   17998  17       683751850   471553        0x0
   0x0  0x90105c8  0x1a005000    14096494  14153835
30.33.1.2    10.10.39.4  4145   30000   18998  17       43858456    30164         0x0
   0x0  0x1a005000 0x1a006600    14096477  14099491
10.10.29.4   30.33.1.2   1479   30000   17998  17       683751850   471553        0x0
   0x0  0x90105c7  0x1a005000    14096476  14153817
10.10.7.4    30.33.1.2   1457   30000   17998  17       683753300   471554        0x0
   0x0  0x90105b1  0x1a005000    14096481  14153822
30.33.1.2    10.10.42.4  4145   30000   18998  17       95289344    65536         0x0
   0x0  0x1a005000 0x1a006600    14112551  14119151
10.10.49.4   30.33.1.2   1499   30000   17998  17       683753300   471554        0x0
   0x0  0x90105db  0x1a005000    14096486  14153827
```

Egress NetFlow

Guidelines and Limitations



Note For verified NetFlow scalability numbers, see Cisco Nexus 9000 Series NX-OS Verified Scalability Guide.

- If both ingress and egress NetFlow are enabled on an interface, ingress NetFlow is prioritized over egress NetFlow and flow information is generated in ingress direction.
- Flows are identified with Egress(1) field in CFLOW packet.
- Egress NetFlow and feature analytics can coexist, but the traffic flows that hit analytics filters are only exported to collectors configured under analytics.
- On modular Cisco Nexus 9500 chassis, traffic comes in and goes out on different line cards. If egress NetFlow is enabled on traffic outgoing interface and not enabled on traffic incoming interface, the egress flow information is captured on ingress interface line card and not captured on egress interface.
- If there are more than one SVI and sub_interface on L2 interface, the ingress interface (if_id) derivation need not be correct for egress NetFlow.
- Egress NetFlow is supported only for IPv4 and Layer 2 traffic. Egress NetFlow does not support IPv6 and multicast traffic.
- Egress NetFlow does not support the following:
  - vrf-id
  - Recording of outgoing control plane traffic
  - Output Interface (NetFlow OIF)
  - MPLS network
- Beginning with Cisco NX-OS Release 10.6(2)F, Nexus 9300-SG2 switches support dropped flow notification feature.
- Beginning with NX-OS Release 10.6(2)F, although ingress NetFlow is supported on Nexus N9336C-SE1, N9396T12C-SE1, N9396Y12C-SE1, N9348Y12C-SE1, N9324C-SE1U, and N9348Y2C6D-SE1U switches, egress NetFlow is not supported on these switches.

Prerequisites

1. Configure the Nexus 9000 switches with egress NetFlow TCAM region.
2. Reload the switch after egress NetFlow TCAM carving.
3. When the switch comes online, enable the egress NetFlow on any L2 or L3 interface under interfaces using the **ip flow monitor <> output or layer2-switched flow monitor <> output** command.

Configure Egress NetFlow

Procedure

Step 1 Enable the NetFlow feature.

Step 2 Perform egr-netflow TCAM carving and reload the switch.

a) Enter the global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
                                switch(config)#
```

b) Enable egress NetFlow TCAM carving using the command [**no**] **hardware access-list tcam region egr-netflow** { **0** | **512** }

Example:

```
switch(config)# hardware access-list tcam region egr-netflow 512
```

The default is disabled. The supported values are 0 and 512 for egr-netflow TCAM carving.

c) Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

d) Reload the switch using the command **reload**

Example:

```
switch(config)# reload
This command will reboot the system. (y/n)? [
n
]
y
```

Step 3 Define a flow record by specifying keys and fields to the flow.

Step 4 Define a flow exporter by specifying the export format, protocol, destination, and other parameters.

Step 5 Define a flow monitor based on the flow record and flow exporter.

Step 6 Apply the flow monitor to a source interface , subinterface, or VLAN interface.

Configuration Example for Egress NetFlow

This example shows how to configure a an egress NetFlow exporter configuration for IPv4:

```
feature netflow
flow exporter ee
 destination 171.70.242.48 use-vrf management
 source mgmt0
 version 9
  template data timeout 20
```

```
flow record rr
 match ipv4 source address
 match ipv4 destination address
 collect counter bytes
 collect counter packets
flow monitor foo
 record rr
 exporter ee
interface Ethernet2/45
 ip flow monitor foo output
 ip address 10.20.1.1/24
 no shutdown
```

Egress NetFlow Verification Commands

To display the egress NetFlow configuration, perform one of the following tasks:

Command	Purpose
show flow cache [**ipv4** | **ce**] [**output**]	Displays information about egress NetFlow IP flows.
show running-config netflow	Displays the egress NetFlow configuration that is currently on your device.

Display Example for Egress NetFlow

The output of the **show flow cache** command for IPv4 displays:

```
switch(config-if-range)# show flow cache
NOTE: Only 10k flows are displayed in XML output
Egress IPV4 Entries
SIP              DIP              BD ID    S-Port   D-Port   Protocol  Byte Count
Packet Count      TCP FLAGS    TOS     if_id        output_if_id       flowStart    flowEnd
     Profile   Ing-VRF
130.1.1.2        162.1.1.1        4119     60       893      6         161082
171               0x0          0x0     0x1a002600   0x1a002800         716994       732532
        4  : NF   21
130.1.1.2        162.1.1.1        4151     60       11013    6         61230
65                0x0          0x0     0x1a006600   0x1a006800         715951       728074
        4  : NF   38
130.1.1.2        162.1.1.1        4145     60       3441     6         162966
173               0x0          0x0     0x1a005a00   0x1a005c00         713085       727941
        4  : NF   35
```

Sampled Netflow

A sampled Netflow (SNF) is an analysis of NetFlow that

- enables sample-based traffic analysis with minimal performance impact,
- supports flexible and configurable sampling rates per flow monitor, and
- integrates seamlessly with existing NetFlow infrastructure and collectors.

NetFlow provides a comprehensive, high-accuracy view of all network conversations, ideal for deep security analysis, while sampled NetFlow gives a statistical overview of traffic, making it better for real-time monitoring of high-speed networks with minimal impact on the device performance.

Concept

Sampled NetFlow allows network administrators to sample and collect information about network traffic flows passing through the switch. SNF focuses on sampled (not all) traffic to reduce performance impact while still providing valuable insights. This sampled traffic data can then be sent to an external collector for analysis. For example, a network engineer can configure SNF to sample traffic on key ports, view flow records, and send this data to security tools to detect unusual patterns or attacks.

Sampled NetFlow uses a sampling mechanism similar to sFlow, using ASIC-level packet sampling. The sampling and export operations are hardware-assisted, ensuring minimal performance overhead. The NetFlow Manager (NFM) manages configuration, sampling rate, and export operations. The sampled NetFlow (SNF) feature provides packet-level traffic sampling for IPv4 and IPv6 flows, allowing administrators to collect representative network data at a configurable sampling rate. The sampled flow records are exported in NetFlow version 9 format for analysis, accounting, and traffic engineering.

Aspects

The three main aspects of SNF include :

- Flow configuration - As part of flow configuration, the focus is on sampling, that is, setting up of the traffic and ports to sample. SNF samples incoming traffic on both physical and logical switch ports and displays this information in tables. For more information, see **Create flow monitor**.
- Flow learning and collection handling - Gathering and managing the sampled traffic flow data is another aspect of SNF. The data is collected for both IPv4 and IPv6 flows. SNF supports flow record collection and formatting. Network administrators can use the information provided by SNF for tasks such as network management, planning, monitoring for security threats such as denial-of-service attacks, accounting, ISP billing, and data analysis.
- Flow export - SNF supports sending the collected data to external collectors for further use. You can export collected traffic flow data using NetFlow version 9 format. You can also export flow data to multiple external collectors at the same time.

Guidelines and Limitations

The guidelines and limitations for sampled NetFlow (SNF) configuration include:

General

- Beginning with 10.6.2(F), a new command is introduced under the existing flow monitor configuration to define the sampling rate.
- For VXLAN tunnel traffic, the outgoing interface is displayed as 0x0.

Supported features

- Sampled NetFlow is supported only on ingress NetFlow.
- Sampled NetFlow supports only two flow monitors at a time, for example, one IPv4 and one IPv6, two IPv4, or two IPv6 flow monitors. Note that the limitation is only on the number of monitors attached to interfaces. You can create more than two monitors but can attach only two of them at a time to interfaces.

- Each flow monitor supports two exporters.

Unsupported features

- sFlow and sampled NetFlow cannot co-exist due to shared session resource limits.
- SNF does not support egress sampled netflow, layer 2 interfaces, and CE monitor.

Hardware-rate limiter

- The default hardware-rate limiter for sampled NetFlow is 40000 pps. The supported range is 0-8,000,000. Use the show hardware rate-limiter netflow command to verify the hardware-rate limiter.

Platform support

- Beginning with 10.6(2)F, the sampled NetFlow feature is supported only on N9364E-SG2-O and N9364E-SG2-Q switches.

General Procedures to Enable NetFlow

The following sections list the procedures that are common to both Ingress and Egress NetFlow when you enable them.

Enable the NetFlow feature

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Enable or disable the NetFlow feature using the command [**no**] **feature netflow**

Example:

```
switch(config)# feature netflow
```

The default is disabled.



Note The Cisco Nexus 9500 platform switches with N9K-T2 EoR do not support NetFlow.

Step 3 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Create a flow record

You can create a flow record and add keys to match on and nonkey fields to collect in the flow.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Create a flow record and enter flow record configuration mode using the command **flow record** *name*

Example:

```
switch(config)# flow record Test
 switch(config-flow-record)#
```

You can enter up to 63 alphanumeric characters for the flow record name.

Step 3 Describe this flow record as a maximum 63-character string using the command **description** *string*

Example:

```
switch(config-flow-record)# description IPv4Flow
```

This is an optional step.

Step 4 Specify a match key using the command **match** *type*

Example:

```
switch(config-flow-record)# match transport destination-port
```

This is an optional step. For more information, see Specify match parameters, on page 459 .

Note The **match transport destination-port** and **match ip protocol** commands are required to export Layer 4 port data.

Step 5 Specify the collection field using the command **collect** *type*

Example:

```
switch(config-flow-record)# collect counter packets
```

This is an optional step. For more information, see Specify collect parameters, on page 460 .

Step 6 Display information about NetFlow flow records using the command **show flow record** [*name*] [*record-name*] { **netflow-original** | **netflow protocol-port** | **netflow** { **ipv4** | **ipv6** } { **original-input** | **original-output** }}

Example:

```
switch(config-flow-record)# show flow record netflow protocol-port
```

This is an optional step. You can enter up to 63 alphanumeric characters for the flow record name.

Step 7 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config-flow-record)# copy running-config startup-config
```

This is an optional step.

Specify match parameters

You must configure at least one of the following match parameters for flow records:

Command	Purpose
match datalink {**mac source-address** | **mac destination-address** | **ethertype** | **vlan**} **Example:** `switch(config-flow-record)# match datalink ethertype`	Specifies the Layer 2 attribute as a key.
match ip {**protocol** | **tos**} **Example:** `switch(config-flow-record)# match ip protocol`	Specifies the IP protocol or ToS fields as keys. **Note** The **match transport destination-port** and **match ip protocol** commands are required to export Layer 4 port data. The data is collected and displayed in the output of the **show hardware flow ip** command but is not collected and exported until you configure both commands.
match ipv4 {**destination address** | **source address**} **Example:** `switch(config-flow-record)# match ipv4 destination address`	Specifies the IPv4 source or destination address as a key.
match ipv6 {**destination address** | **source address** | **flow-label** | **options**} **Example:** `switch(config-flow-record)# match ipv6 flow-label`	Specifies the IPv6 key.

Command	Purpose
match transport {**destination-port** | **source-port**} **Example:** `switch(config-flow-record)# match transport destination-port`	Specifies the transport source or destination port as a key. **Note** The **match transport destination-port** and **match ip protocol** commands are required to export Layer 4 port data. The data is collected and displayed in the output of the **show hardware flow ip** command but is not collected and exported until you configure both commands.

Specify collect parameters

In Cisco/NX-OS NetFlow, “collect parameters” are the configuration options you specify within a flow record that indicate which fields or metrics are captured from a network flow for export and analysis.

You must configure at least one of the following collect parameters for the flow records:

Command	Purpose
collect counter {**bytes** | **packets**} [**long**] **Example:** `switch(config-flow-record)# collect counter packets`	Collects either packet-based or byte counters from the flow. You can optionally specify that 64-bit counters are used.
collect ip version **Example:** `switch(config-flow-record)# collect ip version`	Collects the IP version for the flow.
collect timestamp sys-uptime {**first** | **last**} **Example:** `switch(config-flow-record)# collect timestamp sys-uptime last`	Collects the system up time for the first or last packet in the flow.
collect transport tcp flags **Example:** `switch(config-flow-record)# collect transport tcp flags`	Collects the TCP transport layer flags for the packets in the flow.

Create a flow exporter

The flow exporter configuration defines the export parameters for a flow and specifies reachability information for the remote NetFlow Collector.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Create a flow exporter and enter flow exporter configuration mode using the command **flow exporter** *name*

Example:

```
switch(config)# flow exporter flow-exporter-one
  switch(config-flow-exporter)#
```

You can enter up to 63 alphanumeric characters for the flow exporter name.

Step 3 Set the destination IPv4 or IPv6 address for this flow exporter using the command **destination** { *ipv4-address* | *ipv6-address* } [**use-vrf** *name*]

Example:

```
switch(config-flow-exporter)# destination 192.0.2.1
```

You can optionally configure the VRF to use to reach the NetFlow Collector. You can enter up to 32 alphanumeric characters for the VRF name.

Step 4 Specify the interface to use to reach the NetFlow Collector at the configured destination using the command **source** *interface-type name/port*

Example:

```
switch(config-flow-exporter)# source ethernet 2/1
```

Step 5 Describe this flow exporter using the command **description** *string*

Example:

```
switch(config-flow-exporter)# description exportversion9
```

This is an optional step. You can enter up to 63 alphanumeric characters for the description.

Step 6 Specify the differentiated services codepoint value using the command **dscp** *value* and specify the UDP port to use to reach the NetFlow Collector using the command **transport udp** *port* .

Example:

```
switch(config-flow-exporter)# dscp 0
```

Example:

```
switch(config-flow-exporter)# transport udp 200
```

This is an optional step. The range for differentiated services coodepoint is from 0 to 63. The range for the UDP port is from 0 to 65535.

Note If you do not specify the UDP port, 9995 is selected as the default.

Step 7 Specify the NetFlow export version using the command **version 9**

Example:

```
switch(config-flow-exporter)# version 9
 switch(config-flow-exporter-version-9)#
```

Choose version 9 to enter the flow exporter version 9 configuration submode.

Step 8 Set the flow exporter statistics resend timer using the command **option** { **exporter-stats** | **interface-table** } **timeout** *seconds*

Example:

```
switch(config-flow-exporter-version-9)# option exporter-stats timeout 1200
```

This is an optional step. The range is from 1 to 86400 seconds.

Step 9 Set the template data resend timer using the command **template data timeout** *seconds* and copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config-flow-exporter-version-9)# template data timeout 1200
```

Example:

```
switch(config-flow-exporter-version-9)# copy running-config startup-config
```

This is an optional step. The range for the data resend timer is from 1 to 86400 seconds.

Create Flow Monitor

You can create a flow monitor and associate it with a flow record and a flow exporter. All of the flows that belong to a monitor use the associated flow record to match on the different fields, and the data is exported to the specified flow exporter.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Create a flow monitor and enter flow monitor configuration mode using the command **flow monitor***name*

Example:

```
switch(config)# flow monitor flow-monitor-one
 switch(config-flow-monitor)#
```

You can enter up to 63 alphanumeric characters for the flow monitor name.

Step 3 (Optional) Describe the flow monitor using the command **description***string*

Example:

```
switch(config-flow-monitor)# description IPv4Monitor
```

This is an optional step. You can enter up to 63 alphanumeric characters for the description.

Step 4 Associate a flow exporter with the flow monitor using the command **exporter***name*

Example:

```
switch(config-flow-monitor)# export v9
```

You can enter up to 63 alphanumeric characters for the exporter name.

Step 5 Associate a flow record with the specified flow monitor using the command **record***name* [**netflow-original** | **netflow protocol-port** | **netflow** { **ipv4** | **ipv6** } { **original-input** | **original-output** }]

Example:

```
switch(config-flow-monitor)# record IPv4Flow
```

You can enter up to 63 alphanumeric characters for the record name.



Note **record netflow ipv4 original-input** , **record netflow ipv4 original-output** , and **record netflow layer2-switched input** are not supported in Cisco NX-OS Release 9.3(1). These parameters are not supported on sampled NetFlow.

Step 6 Configure sampling rate using the command **sampling-rate**<*4096-1000000000*>

Example:

```
switch(config)# flow monitor myMonitor
switch(config-flow-monitor)# sampling-rate 5000
```

The default sampling rate is 4096.

Step 7 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config-flow-monitor)# copy running-config startup-config
```

Apply Flow Monitor to an Interface

You can apply a flow monitor to an interface. You can apply a flow monitor in egress direction using the **output** keyword instead of **input** .

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Configure a VLAN interface and enter interface configuration mode using the command **interface vlan** *vlan-id*

Example:

```
switch(config)# interface vlan 10
  switch(config-if)#
```

Step 3 Associate an IPv4, IPv6, or Layer 2-switched flow monitor to the interface for input packets using the command **ip flow monitor** { **ipv4** | **ipv6** | **layer-2-switched** } **input**

Example:

```
switch(config-if)# ip flow monitor ipv4 input
```

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config-if)# copy running-config startup-config
```

This is an optional step.

Additional Procedures for Ingress NetFlow

The following section lists the various other procedures you can apply to Ingress NetFlow.

Configure Bridged NetFlow on VLAN

You can apply a flow monitor to a VLAN in order to gather Layer 3 data over Layer 2 switched packets in a VLAN.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Enter VLAN configuration mode using the command **vlan configuration** *vlan-id*

Example:

```
switch(config)# vlan configuration 30
  switch(config-vlan-config)#
```

The VLAN ID range is from 1 to 3967 or from 4048 to 4093.

Note VLAN configuration mode enables you to configure VLANs independently of their creation, which is required for VTP client support.

Step 3 Associate a flow monitor to the VLAN for input packets using the command { **ip** | **ipv6** } **flow monitor** *name*

Example:

```
switch(config-vlan-config)# ip flow monitor testmonitor
```

You can enter up to 63 alphanumeric characters for the flow monitor name.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config-vlan-config)# copy running-config startup-config
```

This is an optional step.

Configure Layer 2 NetFlow Keys

You can define Layer 2 keys in flexible NetFlow records that you can use to capture flows in Layer 2 interfaces.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Enter flow record configuration mode using the command **flow record** *name*

Example:

```
switch(config)# flow record L2_record
      switch(config-flow-record)#
```

For more information about configuring flow records, see Create a flow record, on page 458 .

Step 3 Specify the Layer 2 attribute as a key using the command **match datalink** { **mac source-address** | **mac destination-address** | **ethertype** | **vlan** }

Example:

```
switch(config-flow-record)# match datalink ethertype
```

Step 4 Exit flow record configuration mode using the command **exit** and enter interface configuration mode using the command **interface** { **ethernet** *slot/port* | **port-channel** *number* }

Example:

```
switch(config-flow-record)# exit
  switch(config)#
```

Example:

```
switch(config)# interface Ethernet 6/3
 switch(config-if#)
```

The interface type can be a physical Ethernet port or a port channel.

Step 5 Change the interface to a Layer 2 physical interface using the command **switchport**

Example:

```
switch(config-if)# switchport
```

For information on configuring switch ports, see the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide .

Step 6 Force MAC classification of packets using the command **mac packet-classify**

Example:

```
switch(config-if)# mac packet-classify
```

For more information on using this command, see the Cisco Nexus 9000 Series NX-OS Security Configuration Guide .

Note You must use this command to capture flows.

Step 7 Associate a flow monitor to the switch port input packets using the command **layer2-switched flow monitor** *flow-name* **input**

Example:

```
switch(config-if)# layer2-switched flow monitor L2_monitor input
```

You can enter up to 63 alphanumeric characters for the flow monitor name.

Step 8 Display information about the Layer 2 NetFlow default record using the command **show flow record netflow layer2-switched input**

Example:

```
switch(config-if)# show flow record netflow layer2-switched input
```

This is an optional step.

Step 9 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config-if)# copy running-config startup-config
```

This is an optional step.

Configure Layer 3 NetFlow on Layer 2 Interfaces

You can define Layer 3 flow monitors on Layer 2 interfaces to capture Layer 3 flow information on Layer 2 interfaces.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Enter flow record configuration mode using the command **flow record** *name*

Example:

```
switch(config)# flow record L3_record
 switch(config-flow-record)#
```

For more information about configuring flow records, see Create a flow record, on page 458 .

Step 3 Enter interface configuration mode using the command **interface** { **ethernet** *slot/port* | **port-channel** *number* }

Example:

```
switch(config)# interface Ethernet 6/3
  switch(config-if#)
```

The interface type can be a physical Ethernet port or a port channel.

Step 4 Change the interface to a Layer 2 mode using the command **switchport**

Example:

```
switch(config-if)# switchport
```

For information on configuring switch ports, see the Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide .

Step 5 Associate an IPv4 flow monitor to the switch port input packets using the command **ip flow monitor** *flow-name* **input**

Example:

```
switch(config-if)# ip flow monitor v41 input
```

You can enter up to 63 alphanumeric characters for the flow monitor name.

Step 6 Associate an IPv6 flow monitor to the switch port input packets using the command **ipv6 flow monitor** *flow-name* **input**

Example:

```
switch(config-if)# ipv6 flow monitor v61 input
```

You can enter up to 63 alphanumeric characters for the flow monitor name.

Step 7 Copy the running configuration to the startup configuration using the commnd **copy running-config startup-config**

Example:

```
switch(config-if)# copy running-config startup-config
```

This is an optional step.

Configure NetFlow Timeouts

You can optionally configure global NetFlow timeouts that apply to all flows in the system.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Set the flush timeout value in seconds uisng the command **flow timeout** *seconds*

Example:

```
switch(config)# flow timeout 30
```

The range is from 5 to 60 seconds. The default value is 10 seconds.

Step 3 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Additional Procedures for Egress NetFlow

You can define Layer 3 flow monitors on Layer 2 interfaces to capture Layer 3 flow information on Layer 2 interfaces.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Enter flow record configuration mode using the command **flow record** *name*

Example:

```
switch(config)# flow record L3_record
  switch(config-flow-record)#
```

For more information about configuring flow records, see Create a flow record, on page 458 .

Step 3 Enter interface configuration mode using the command **interface** { **ethernet** *slot/port* | **port-channel** *number* }

Example:

```
switch(config)# interface Ethernet 6/3
  switch(config-if#)
```

The interface type can be a physical Ethernet port or a port channel.

Step 4 Change the interface to a Layer 2 mode using the command **switchport**

Example:

```
switch(config-if)# switchport
```

For information on configuring switch ports, see Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide .

Step 5 Associate an IPv4 flow monitor to the switch port input or output packets using the commands **ip flow monitor** *flow-name* **input** | **output**

or

layer2-switched flow monitor *flow-name* **input** | **output**

Example:

```
switch(config-if)# ip flow monitor v41 input

switch(config-if)# layer2-switched flow monitor v41 output
```

You can enter up to 63 alphanumeric characters for the flow monitor name.

Step 6 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config-if)# copy running-config startup-config
```

This is an optional step.

Drop Reports

Beginning with Cisco NX-OS 10.5(2), users can utilize a new troubleshooting feature called the drop report. This can be used to help quickly narrow down troubleshooting like intermittent packet loss or congestion on single flow. This feature is supported on Cisco Nexus 9300- FX3/GX/GX2/HX platform switches. This feature supports the following drops:

- Policer Drops

- Forward Drops
- IDS Drops
- Buffer Drop
- RPF Drops

Guidelines and Limitations

This drop report has the following guidelines and limitations:

- Feature NetFlow is automatically enabled when users configure the drop report feature.
- A maximum of 10 flows matching the specified source and destination IPs are supported. To capture a different flow with specified source and destination IPs, the filter must be cleared and re-applied to restart report capturing.
- The drop report filter stays active for 10 minutes.
- The drop report feature is not supported if the feature analytics is already configured.
- Only single unique flow with 2-tuple(sip/dip) and best effort 5-tuple is supported at a time.
- This feature has no ISSU, ISSD, and reload support.

Configure Drop Report

You need to configure this feature by entering into the drop-report mode.

Procedure

Step 1 Enter drop-remote mode using the command **drop-report**

Example:

```
switch# drop-report
switch(drop-report)#
```

Step 2 Enable the NetFlow feature using the command **start**

Example:

```
sswitch(drop-report)# start
Feature netflow enabled by drop report tool
```

Step 3 Configure filter to assign parameters to the Access Control List (ACL) using the command **filter {ipv4 [src-ip | dst-ip] | ipv6 [src-ip | dst-ip]}**

Example:

```
switch(drop-report)# filter ipv6 src-ip 100::1 dst-ip 200::2
                      switch(drop-report)# filter ipv4 src-ip 192.168.1.2  dst-ip 192.168.1.1
```

Step 4 Shows drop information using the command **report**

Example:

```
switch(drop-report)# report
```

Step 5 Removes ACL and stop recording using the command **clear**

Example:

```
switch(drop-report)# clear
                      CS drop filter removed
```

Step 6 Remove drop report feature using the command **stop**

Example:

```
switch(drop-report)# stop
```

Verify Drop Report Configuration

Use the following command to see the drop-report result:

For IPv4

```
switch# drop-report
switch(drop-report)# start
Feature netflow enabled by drop report tool
switch(drop-report)# filter ipv4 src-ip 10.0.0.9 dst-ip 10.0.0.10
switch(drop-report)# report
Filter Status: Applied    Filter Type  : IPv4
Dst IP: 10.0.0.10 Src IP: 10.0.0.9 Protocol:  Dst Port:  Src Port:

IPV4 Entries
SIP             DIP             BDId  SrcP  DstP  Prot  Byte Count       Packet Count
Ing Interface   Fwd Rpf Pol Ids Acl Buf
10.0.0.9        10.0.0.10       4112  49280 3784  17    70               1
Ethernet1/8     0   0   0   0   0   0


Fwd: Forward Drop Rpf:RPF Fail Pol:Policer Drop Ids: Ids Drop Acl: Acl Drop Buf: Buffer
Drop
```

For IPv6

```
switch(drop-report)# report
Filter Status: Applied    Filter Type  : IPv6
Dst IPv6: 2000:1::130:1:33:10 Src IPv6: 2000:1::22:1:1:10 Protocol:  Dst Port:  Src Port:

IPV6 Entries
SIP               DIP                 BDId  SrcP  DstP  Prot  Byte Count       Packet Count
     Ing Interface   Fwd Rpf Pol Ids Acl Buf
2000:1::22:1:1:10 2000:1::130:1:33:10 4100  300   400   17    76494000         69540
      Ethernet1/1     0   0   0   0   0   0

2000:1::22:1:1:10 2000:1::130:1:33:10 4100  306   400   17    77632500         70575
      Ethernet1/1     0   0   0   0   0   0

2000:1::22:1:1:10 2000:1::130:1:33:10 4100  303   400   17    76320200         69382
      Ethernet1/1     0   0   0   0   0   0
```

```
2000:1::22:1:1:10 2000:1::130:1:33:10 4100  304   400   17    77745800         70678
      Ethernet1/1     0   0   0   0   0   0

2000:1::22:1:1:10 2000:1::130:1:33:10 4100  308   400   17    77867900         70789
      Ethernet1/1     0   0   0   0   0   0

2000:1::22:1:1:10 2000:1::130:1:33:10 4100  301   400   17    76458800         69508
      Ethernet1/1     0   0   0   0   0   0

2000:1::22:1:1:10 2000:1::130:1:33:10 4100  307   400   17    77558800         70508
      Ethernet1/1     1   0   0   1   1   0

2000:1::22:1:1:10 2000:1::130:1:33:10 4100  309   400   17    77943800         70858
      Ethernet1/1     0   0   0   0   0   0

2000:1::22:1:1:10 2000:1::130:1:33:10 4100  302   400   17    76373000         69430
      Ethernet1/1     0   0   0   0   0   0

2000:1::22:1:1:10 2000:1::130:1:33:10 4100  305   400   17    77706200         70642
      Ethernet1/1     0   0   0   0   0   0

Fwd: Forward Drop Rpf:RPF Fail Pol:Policer Drop Ids: Ids Drop Acl: Acl Drop Buf: Buffer
Drop

switch(drop-report)# clear
CS drop filter removed

switch(drop-report)# report
Filter Status: No Filter Applied
```

Dropped flow notification

Dropped flow notification is a feature that

- allows network administrators to capture packets that are otherwise dropped due to forwarding issues or insufficient buffer space in the queue,
- achieves this by copying the drop packets to CPU for the configured drop events, and
- exporting them to collector in version 9 format indicating the flow details and the reason for the drop, thus assisting in identifying the cause of packet loss.

Note

- The export limit based on the configurable timer is 64,000.
- The packets are rate limited to CPU so the dropped packet count is approximate.

Configure Dropped Flow Notification

Perform these steps to configure dropped flow notification.

Procedure

Step 1 Enable the NetFlow feature.

Step 2 Enable event export interval using the **flow event export-interval** <*interval*> command to specify the rate at which the data is exported. The range of export-interval is 0-60 seconds and default value is 10 seconds.

Example:

```
switch(config)# flow event export-interval 50
```

Step 3 Create a flow exporter.

Step 4 Configure a UDP port using the **events transport udp** <*udp-port*> command for exporting event data. The range of UDP port is 1-65535.

Example:

```
switch(config-flow-exporter)# events transport udp 9001
```

Step 5 Create a flow record.

Step 6 Create flow monitor.

Step 7 Configure events to be captured using the **flow event** <*event_name*> command.

Example:

```
switch(config-flow-monitor)# flow event ee
```

a) Use the **capture buffer-drops**, **capture fwd-drops**, **capture acl-drops**, or all three event configuration commands to define the packet drops that should be monitored.

Example:

```
switch(config-flow-event)# capture buffer-drops

switch(config-flow-event)# capture acl-drops

switch(config-flow-event)# capture fwd-drops
```

Step 8 Enter the **flow system config** command in global configuration mode to define system-level flow event settings.

Example:

```
switch(config-flow-event)# flow system config
```

a) Enter the **monitor** <*mon_name*> **input** command to associate an existing NetFlow monitor with the system's input traffic.

Example:

```
switch(config-flow-system)# monitor mon_ipv4 input
```

b) Enter the **event** <*event_name*> command to apply a pre-configured event definition to the system-wide flow event configuration.

Example:

```
switch(config-flow-system)# event ee
```

Sample configuration for dropped flow notification

Here is an example that displays the complete configuration for dropped flow notification.

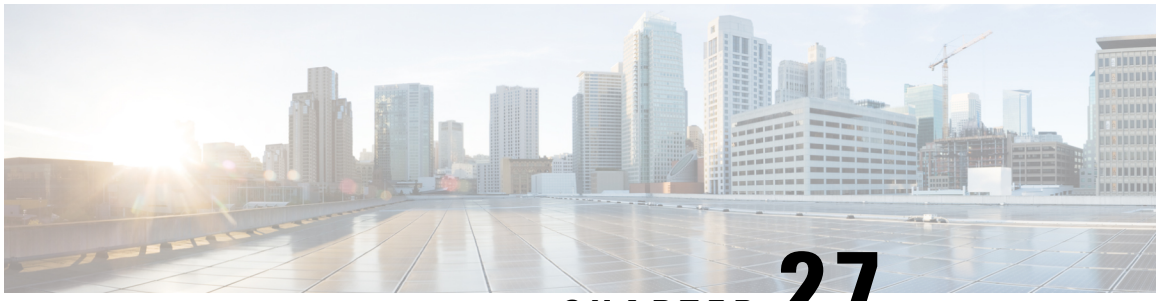
```
switch(config)# feature netflow
switch(config)# flow event export-interval 50
switch(config)# flow exporter exp_2
switch(config-flow-exporter)# destination 50.50.50.53
switch(config-flow-exporter)# transport udp 9996
switch(config-flow-exporter)# source Ethernet1/7
switch(config-flow-exporter)# events transport udp 9001
switch(config-flow-exporter)# version 9
switch(config-flow-exporter-version-9)# flow record rec_ipv4
switch(config-flow-record)# match ipv4 source address
switch(config-flow-record)# match ipv4 destination address
switch(config-flow-record)# match ip protocol
switch(config-flow-record)# match ip tos
switch(config-flow-record)# match transport source-port
switch(config-flow-record)# match transport destination-port
switch(config-flow-record)# flow monitor mon_ipv4
switch(config-flow-monitor)# record rec_ipv4
switch(config-flow-monitor)# exporter exp_2
switch(config-flow-monitor)# flow event ee
switch(config-flow-event)# capture buffer-drops
switch(config-flow-event)# capture acl-drops
switch(config-flow-event)# capture fwd-drops
switch(config-flow-event)# flow system config
switch(config-flow-system)# monitor mon_ipv4 input
switch(config-flow-system)# event ee
switch(config-flow-system)#
```

Verification commands for dropped flow notification

Based on your purpose, use the show commands for dropped flow notification configuration mentioned in the table.

Table 34: Verification commands for dropped flow notification

Command	Purpose
show flow event	Displays information about the drop events.
show flow event cache [**ipv4** | **ipv6**]	Displays information about dropped flows.
show flow system	Displays information about the system configuration.
show flow event export-interval	Displays the interval at which dropped flows are exported.



CHAPTER 27

Mixed Mode

This chapter describes how to configure the Mixed Mode (Analytics and NetFlow) feature on NX-OS devices.

This chapter contains the following sections:

- Mixed Mode, on page 475
- Guidelines and limitations for Mixed Mode, on page 475
- Mixed mode use cases, on page 476
- Commands for verifying the Mixed Mode configuration, on page 479
- Display example for Mixed Mode, on page 480

Mixed Mode

You can configure NetFlow and Analytics features on a switch, so both the features coexist and utilize the standard Version 9 export from the Central Processing Unit (CPU). This mode, in which both the features coexist, is called the Mixed Mode.

Note Until NX-OS Release 10.2(3)F, standard Version 9 export was supported only for NetFlow flow records. Beginning from NX-OS Release 10.2(3)F, standard Version 9 export is supported for Analytics also. However, NetFlow and Analytics features are mutually exclusive.

Guidelines and limitations for Mixed Mode

The guidelines and limitations applicable to Mixed Mode are:

- • L2 flow monitor is not supported.
  - VRF filter is not supported.
  - ND ISSU is not supported.
  - The IPv4 and IPv6 profiles are as follows:
    - IP flow monitor: 28
    - IPv6 flow monitor: 26

- Analytics record configuration must be a superset of all the record parameters.
- Configure system monitor before configuring any system filter/interface filter configurations.
- Unconfigure system filter/interface filter configurations, before unconfiguring system monitor.
- In Mixed mode, two NetFlow records are exported for AN flow on EOR.

- Interface based FT is not supported for tunnel traffic flows such as MPLS, VXLAN, and GRE.
- Beginning with NX-OS Release 10.3(3)F, ingress_vrf_id is supported for the NetFlow and Analytics features on all Nexus 9000 switches.

  The ingress vrf-id is captured, shown in **show flow cache** and sent to NetFlow collector.

  When Layer 3 NetFlow is configured on a Layer 2 interface and the traffic is sent, and then the **show flow cache** command output displays the value of Ingress_VRF_ID as zero.
- Beginning with NX-OS Release 10.3(3)F, the NetFlow mixed mode is enabled by default. This reduces the TCAM space assigned to the Analytics feature from a maximum of 512 entries to a maximum of 256 entries.
- Beginning with NX-OS Release 10.3(3)F, flow record is seen when it is defined in system filter, but not defined in interface filter unlike in earlier releases. In the earlier releases, if the interface filter is configured, the flow record was seen only if it was defined in the interface filter.
- Beginning with NX-OS Release 10.5(1)F, ND ISSU is supported.

Mixed mode use cases

The three possible use cases or scenarios for Mixed mode are:

- switches already deployed with feature analytics
- switches already deployed with feature netflow, and
- switches that have neither feature configured.

Mixed mode can be configured only from NetFlow mode. In a scenario where the switches already have feature analytics enabled, unconfigure analytics first, configure feature netflow, and then transition to Mixed mode.

After configuring the mixed mode, use the standard Version 9 format to export both NetFlow and Analytics flow records from the CPU to the respective collectors.

Note Analytics data is a superset of NetFlow data. The additional analytics flow data such as flow latency, traffic burst data, payload length, TCP flags, IP flags, and packet disposition flags is communicated through Vendor Specific Fields (VSF).

Switches already deployed with feature analytics

This section provides information for the use case or scenario where the switch is already deployed with feature analytics.

Unconfigure or save feature analytics configuration and perform the steps indicated in Switches that have neither features configured. Note that transition to Mixed mode is not possible to or from Analytics mode.

Switches already deployed with feature netflow

The steps mentioned in this use case is for switches that already have feature netflow deployed on them.

Procedure

Configure feature analytics based on this sample configuration.

Example:

```
feature analytics
           analytics
           flow filter telemetryFP
           ipv4 telemetryIpv4Acl
           ipv6 telemetryIpv6Acl
           flow exporter e11
           destination 10.10.20.21 v9
           transport udp 1100
           events transport udp 55
           source Ethernet1/42
           flow exporter e12
           destination 10.10.20.21 v9
           transport udp 9200
           events transport udp 555
           source Ethernet1/42
           flow record fte-record
           match ip source address
           match ip destination address
           match ip protocol
           match transport source-port
           match transport destination-port
           collect counter packets
           collect timestamp sys-uptime first
           collect timestamp sys-uptime last
           flow monitor m1
           record fte-record
           exporter-bucket-id 1 0 4095
           exporter e11
           flow monitor m2
           record fte-record
           exporter-bucket-id 1 0 2000
           exporter e11
           exporter-bucket-id 2 2001 4095
           exporter e12
           flow profile telemetryProf
           collect interval 1000
           source port 1001
           flow event fte-event1
           group drop-events
           capture buffer-drops
           capture acl-drops
```

```
capture fwd-drops
group packet-events
capture tos 50
capture ttl 50
flow system config
exporter-id 4
monitor m1 input
profile telemetryProf
event fte-event1
filter telemetryFP
```

Switches that have neither features configured

This use case is for switches that have neither NetFlow nor Analytics features configured.

Procedure

Configure feature netflow and then perform either the steps mentioned in Switches already deployed with feature NetFlow or from the sample configuration provided here.

Example:

```
feature netflow
        feature analytics
        flow exporter e1
        destination 10.10.20.21
        transport udp 100
        source Ethernet1/42
        version 9
        flow record r4
        match ipv4 source address
        match ipv4 destination address
        match ip protocol
        match transport source-port
        match transport destination-port
        collect counter bytes
        collect counter packets
        collect timestamp sys-uptime first
        collect timestamp sys-uptime last
        flow record r6
        match ip protocol
        match transport source-port
        match transport destination-port
        match ipv6 source address
        match ipv6 destination address
        collect counter bytes
        collect counter packets
        collect timestamp sys-uptime first
        collect timestamp sys-uptime last
        flow monitor m41
        record r4
        exporter e1
        flow monitor m6
        record r6
        exporter e1
        analytics
```

```
flow filter telemetryFP
ipv4 telemetryIpv4Acl
ipv6 telemetryIpv6Acl
flow exporter e11
destination 10.10.20.21 v9
transport udp 1100
events transport udp 55
source Ethernet1/42
flow exporter e12
destination 10.10.20.21 v9
transport udp 9200
events transport udp 555
source Ethernet1/42
flow record fte-record
match ip source address
match ip destination address
match ip protocol
match transport source-port
match transport destination-port
collect counter packets
collect timestamp sys-uptime first
collect timestamp sys-uptime last
flow monitor m1
record fte-record
exporter-bucket-id 1 0 4095
exporter e11
flow monitor m2
record fte-record
exporter-bucket-id 1 0 2000
exporter e11
exporter-bucket-id 2 2001 4095
exporter e12
flow profile telemetryProf
collect interval 1000
source port 1001
flow event fte-event1
group drop-events
capture buffer-drops
capture acl-drops
capture fwd-drops
group packet-events
capture tos 50
capture ttl 50
flow system config
exporter-id 4
monitor m1 input
profile telemetryProf
event fte-event1
filter telemetryFP
interface Ethernet1/42
ip flow monitor m41 input
ipv6 flow monitor m6 input
```

Commands for verifying the Mixed Mode configuration

Use any of the show commands listed in the table to verify the required mixed mode configuration.

Command	Purpose
show flow cache [**ipv4** | **ipv6**]	Displays information about NetFlow IP flows. **Note** Beginning with NX-OS Release 10.3(3)F, this command output also displays Ingress VRF ID. The ingress vrf-id is captured, shown in show flow cache and sent to NetFlow collector.
show flow exporter [*name*]	Displays information about NetFlow and Analytics flow exporters and statistics. You can enter up to 63 alphanumeric characters for the flow exporter name.
show flow interface [*interface-type slot/port*]	Displays information about NetFlow and Analytics interfaces.
show flow record [*name*]	Displays information about NetFlow and Analytics flow records. You can enter up to 63 alphanumeric characters for the flow record name.
show running-config [**netflow** | **analytics**]	Displays the coexisting NetFlow and Analytics configuration that is currently on your device.
show flow monitor	Displays the NetFlow and Analytics monitor configuration.
show flow system	Displays information about the Analytics system configuration.
show flow filter	Displays information about Analytics filters.
show flow profile	Displays information about the Analytics profile.
show flow event	Displays information about the Analytics events.

Display example for Mixed Mode

This section provides sample outputs of the **show flow cache** command.

Note

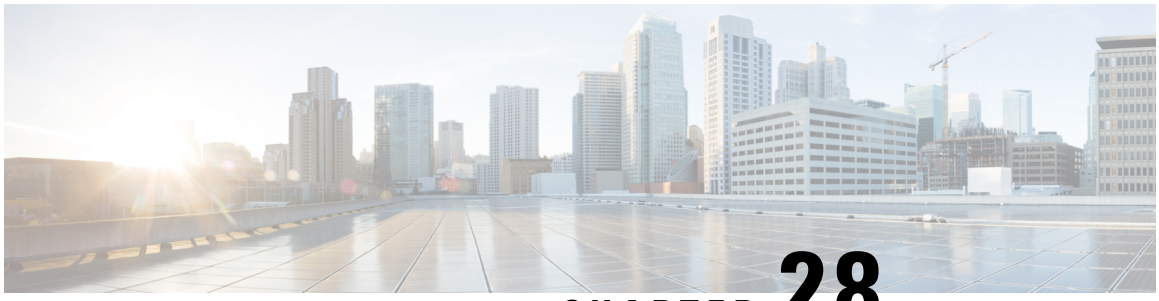
- Only 10,000 flows are displayed in XML output.
- When Layer 3 NetFlow is configured on a Layer 2 interface and the traffic is sent, and then the **show flow cache** command is run, the output displays the value of Ing-VRF as zero.

show flow cache

```
Ingress IPV4 Entries
SIP             DIP          BD ID    S-Port   D-Port   Protocol  Byte Count   Packet Count
  TCP FLAGS    TOS     if_id       flowStart     flowEnd      Profile    Ing-VRF
```

```
192.0.2.2        192.0.2.1    1671     0        0        89        480          8
    0x0          0xc0     0x1a004400  2938966       2976728      5  : NF    0
192.0.2.2        224.0.0.13   1672     0        0        103       144          2
    0x0          0xc0     0x1a004400  2941719       2969951      5  : NF    0
192.0.2.2        224.0.0.13   1675     0        0        103       72           1
    0x0          0xc0     0x1a004400  2961417       2961667      5  : NF    0
192.0.2.2        224.0.0.5    1675     0        0        89        340          5
    0x0          0xc0     0x1a004400  2943341       2979400      5  : NF    0
192.0.2.2        192.0.2.1    1671     2048     0        1         3612         43
    0x0          0x0      0x1a004400  2938188       2980184      5  : NF    0

Ingress IPV6 Entries
SIP                         DIP       BD ID    S-Port  D-Port  Protocol  Byte Count Packet
 Count  TCP FLAGS Flow Label  if_id       flowStart  flowEnd    Ing-VRF
fe80::822d:bfff:fe81:e415   ff02::5   4147     0       0       89        490        5
        0x0       0x0         0x1a003400  11217548   11254367   1
```



CHAPTER 28

Configuring Traffic Analytics

This chapter describes how to configure the Traffic Analytics feature on Cisco NX-OS devices.

- Traffic Analytics, on page 483
- Guidelines and limitations for Traffic Analytics, on page 488
- Configure Traffic Analytics, on page 490
- Example for Traffic Analytics interface filter and VRF filter, on page 491
- Example for Traffic Analytics, on page 493

Traffic Analytics

Traffic Analytics is a feature that

- provides the ability to identify services offered by servers behind a switch, aggregates analytics data, and exports summarized flow records for analysis
- distinguishes between servers and clients using TCP flags (SYN and SYN ACK) in a three-way handshake
- collapses multiple TCP session data traffic into a single record in the show flow cache database and exports it to the collector; during aggregation, the source port of TCP is set to 0
- supports faster export cadence for troubleshoot flows, and
- supports traffic analytics interface filter and VRF filter.

A flow is defined by the source interface, protocol, source IP address, source port, destination IP address, and destination port values. If traffic analytics is enabled, the flows of TCP sessions are aggregated based on source IP address (SIP), destination IP address (DIP), source port (SP) for server to client traffic and SIP, DIP, destination port (DP) for client to server traffic.

Aging of traffic database entries

The traffic database entries will be monitored every 24 hours using a timer. If there is no traffic hitting a database entry, then within 24 to 48 hours that traffic database entry will be deleted. By default the size of the database is 5000.

Troubleshooting rules

The troubleshooting rules are used to debug a flow by programming an analytics ACL filter. These rules take precedence over the traffic analytics rules and can be used for capturing specific flow. Troubleshooting rules might result in two entries in the flow cache.

Troubleshooting rules should be used only for specific flows preferably host for short duration only.

Faster export cadence for troubleshoot flow records

Currently, the flow records and troubleshoot records are exported at a fixed interval of one minute. A new **filter export-interval** command is introduced. This command facilitates the export of troubleshoot records at a faster interval by utilizing a dedicated hash database.

This configuration can be applied only if traffic analytics is enabled, and a filter is set up within the flow system settings. For more information on **filter export-interval** command, see Example for Traffic Analytics, on page 493.

UDP port support and configuration

UDP port support in Traffic Analytics allows masking of exported flows based on configured UDP ports.

- When UDP ports are configured, flows are masked in the TA DB and NFM flow cache.
- If the destination port matches, the source port is masked, and if the source port matches, the destination port is masked.
- If UDP port is not configured, the current functionality is not impacted.



Note First enter the NetFlow entry, and then make the TA entries.

UDP port configuration

Configure UDP port(s) for masking exported flows, use the [**no**] **udp-port** *port-range* command under the flow traffic-analytics submode (under analytics).

- The UDP port must be in the range of 1 to 65565.
- Ports can be entered in a comma-separated and/or range-based format (for example: `2000-3000, 400, 500`).

When the number of ports in the input exceeds the maximum number of ports that can be displayed in a single line command, they are spilled over to a new configuration line as shown in this example.

```
analytics
  flow traffic-analytics
    udp-port
5,40,50,102,104,106,108,110,112,204,206,302,304,306,308,310,312,404,406,502,504,506,508,510,512,604,606,608,610,612,704,706,708,710,712,804,806,808,810,812,904,906,908,910,912,1004,1006,1008,1010,1012,1104,1106,1108,1110,1112,1204,1206,1208,1210,1212

    udp-port
1124,1126,1128,1130,1132,1134,1136,1138,1140,1142,1144,1146,1148,1150,1152,1154,1156,1158,1160,1162,1164,1166,1168,1170,1172,1174,1176,1178,1180,1182,1184,1186,1188,1190,1192,1194,1196,1198,1200,2000-3000,3002,3004,3006
```

Traffic Analytics interface filter and VRF filter

The Traffic Analytics (TA) feature is enhanced to offer more granular support to capture TCP flows using filter configuration at both the interface and VRF levels, similar to the existing FT interface configuration.

Under this TA filter configuration, you can achieve the following:

- Configure an IP address that is required for monitoring and use keywords such as
  - **permit** for IP address that requires monitoring,
  - **deny** to avoid the flow being collected, and
  - **ft-collapse** to integrate flows into a single service.

Note The **ft-collapse** keyword is not used for **troubleshoot filters**.

- Configure the VRF filter across all interfaces in a given VRF.
- Provide permit subnet rules for TCP packets (TCP SYN, SYN ACK, and without any TCP flag).
- For general TCP packets (without SYN or SYN ACK) which are considered for profile 31, the TCS flows forwarded to the collector can be stopped using the **show flow cache** command.
- The **output** option is introduced for flow filter to be applied in the egress direction only.
- Both IPv4 and IPv6 access-list are supported under the filter.

For more information on TA interface filter and VRF filter, see Example for Traffic Analytics interface filter and VRF filter, on page 491.

Interface Traffic Analytics

Interface Traffic Analytics is a network analytics feature that provides

- granular flow control for ingress and egress interfaces, enabling specific actions on traffic that comes in and goes out of interfaces
- filters can be applied at both the interface and VRF level, similar to FT interface configuration, and
- removal of Traffic Analytics is not allowed if it is enabled at the interface level.

Granular flow control for Traffic Analytics is provided for

- ingress interface and
- egress interface.

This feature provides an ability to take specific action on traffic that comes in and goes out of interfaces.

Note Removal of TA is not allowed if it is enabled at interface level.

Granular control for TA feature is provided on ingress and egress interfaces by allowing filters at interface and VRF level, like FT interface configuration.

The table displays the interfaces that are supported on ingress and egress interfaces through releases.

Table 35: Supported interfaces for ingress and egress Traffic Analytics

Interface	Ingress Support	Egress Support
SVI interface	From Release 10.5(2)F	From Release 10.5(3)F
sub interface	From Release 10.5(3)F	From Release 10.5(3)F
port-channel interface	From Release 10.5(2)F	From Release 10.5(3)F
VRF interface	From Release 10.5(2)F	From Release 10.5(3)F
VNI interface	From Release 10.5(3)F	From Release 10.5(3)F

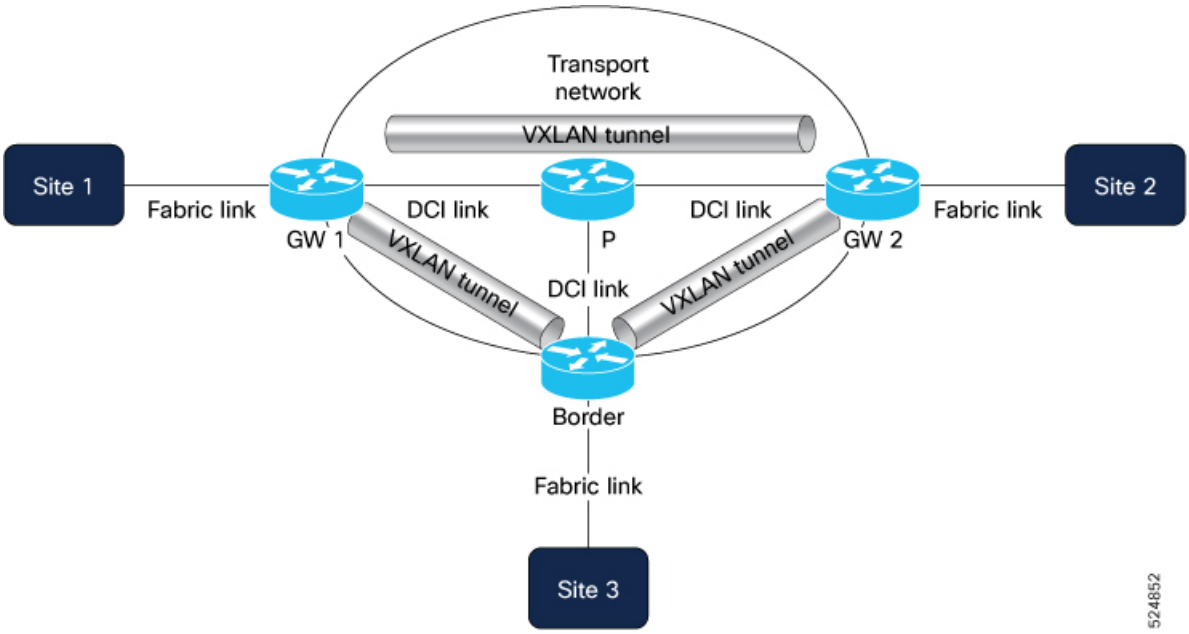
VNI interface

Flow filters can be applied under Layer 3 VNI interfaces in a VXLAN fabric like any other interface filters. The flow filters are supported in both ingress and egress directions. The filters can be either IPv4 or IPv6 filters.

The limitations for TA on VNI interface include:

- Bridged traffic or Layer 2 forwarded traffic cannot be filtered using flow filters applied under L3VNI interface in BGW of a VXLAN fabric.
- A deny flow filter in one direction blocks the traffic in the opposite direction too as VNI interfaces are not direction aware in BGWs of a VXLAN fabric.

Figure 9: Traffic Analytics in VNI interface



The image depicts what happens during Traffic Analytics in the VNI interface in ingress and egress directions.

- Ingress - All traffic coming from DCI link gets decapsulated using VTEP and goes through the VNI interface where policy is applied. It is then forwarded to either fabric or host interface.
- Egress – All traffic coming from either fabric or host interface goes through VNI interface where policy is applied and gets encapsulated using VTEP. It is then forwarded to the DCI link.

Example configuration for VNI interface.

```
vrf context TENANT-VRF
vni 70000 l3

interface nve1
  member vni 70000 associate-vrf


interface vni70000
  flow filter v4_vni_filter_input
  flow filter v4_vni_filter_output output
```

ECN detection for Traffic Analytics

Explicit Congestion Notification (ECN) is a mechanism that provides

- Enhanced network management—Accurate detection of ECN bits provides administrators with the necessary information to effectively manage congestion, such as rerouting traffic or adjusting bandwidth.
- Optimized quality of service—By focusing on CE notifications, this feature helps keep real-time applications running smoothly, allowing for proactive management of congestion.
- Better troubleshooting—Monitoring ECN bits provides detailed insights into the network's health, aiding in quick fixes and long-term planning.

Explicit Congestion Notification (ECN) helps network devices signal congestion without losing packets. It focuses on the CE (Congestion Experienced) notification, which indicates that a packet has encountered congestion on its path. The enhancement in Traffic Analytics allows the system to find and report ECN bits in the IP header. This feature is designed for use with switches managed by Network Insights, where records are exported to Network Insights Resources (NIR) for consumption and further analysis.

This feature is crucial for closely monitoring and managing congestion across network traffic. It is particularly beneficial for real-time applications, such as VoIP calls and video streaming, where maintaining consistent quality is vital. By focusing on CE notifications and leveraging Network Insights Resources (NIR), network managers gain insights into congestion patterns, helping maintain performance stability in environments sensitive to delays.

How ECN Detection Works

These stages describe how the Traffic Analytics system detects and reports ECN bits in IP traffic:

1. The Traffic Analytics system continuously monitors IP traffic.
2. For each packet, the system examines the IP header to detect ECN bits, specifically looking for the CE (Congestion Experienced) notification.
3. When ECN bits are detected, the system records this information, identifying instances of congestion.

4. The collected data is used to generate reports or alerts for network administrators, highlighting congestion areas, and is further analyzed using NIR.

This process ensures that network administrators receive timely and accurate information about congestion in IP traffic, enabling effective management and optimization of network performance.

Disable global Traffic Analytics

Procedure

Configure **mode interface** under **flow traffic-analytics** to disable global Traffic Analytics.

Example:

```
switch(config)# analytics
switch(config)# flow traffic-analytics
switch(config)# mode interface
```

Use the **no** form of the **mode interface** command to disable the interface mode.

Guidelines and limitations for Traffic Analytics

The guidelines and limitations are applicable to Traffic Analytics are:

- If the Traffic Analytics feature is enabled, other than TCP all other IP protocols get 3 tuple information.
- The Traffic Analytics feature is supported only on Mixed mode in standalone devices.
- Before enabling the traffic analytics feature, ensure to remove the flow filters else an error message is displayed.
- If the traffic analytics database size is reduced, new entries happen only after removing the old entries.

  When a system flow filter is configured, the traffic flow behavior is as follows:

  - If a traffic analytics database has information, two flows are seen in the cache.
  - If a traffic analytics database does not have information, only one flow is seen in the cache.

- When both NetFlow and traffic analytics are enabled, the profiles used for both functions in a scaled NetFlow configuration are:

  - 29-31 until NX-OS Release 10.5(2)F
  - 26-31 from NX-OS Release 10.5(3)F

  When neighbor discovery or special packets hit these profiles, it is not possible to distinguish whether the record created is for traffic analytics or NetFlow. Consequently, the record gets processed twice, resulting in the appearance of two packets with one AN profile.

- Netflow and Flow Telemetry are not supported in N9K-C9364C-H1 platform SFP+ ports, Ethernet1/65, and Ethernet1/66.
- Beginning with NX-OS Release 10.5(2)F, ingress traffic analytics is supported.
- Beginning with NX-OS Release 10.5(3)F, the traffic analytics features supported are
  - egress traffic analytics,
  - in ingress traffic analytics:
    - sub-interface,
    - VNI level interfaces, and
    - collapse action.
  - explicit congestion notification for flows,
  - global traffic analytics is also supported on Nexus 9500 switches with GX and FX line cards, and
  - mode interface disables global traffic analytics only on 9300-FX3, GX, GX2, H2R, and H1 switches.

Platform support

The table lists the supported platforms for TA features through releases.

Features	Platforms	Release
Support for Traffic Analytics	9300-FX, FX2, FX3, GX, GX2, and H2R	10.4(2)F
Support for Traffic Analytics	9300-H1	10.4(3)F
Support for Traffic Analytics	9300-H2R and H1	10.4(4)M
Ingress Traffic Analytics	9300-FX, FX2, FX3, GX, GX2, H2R, and H1	10.5(2)F
Egress Traffic Analytics	9300-FX3, GX, GX2, H2R, and H1	10.5(3)F
Global Traffic Analytics	9500 with GX and FX line cards	10.5(3)F
Egress Traffic Analytics	9300-FX and FX2	10.6(1)F



Note For more information about supported platforms for features through releases, refer to Nexus Switch Platform Support Matrix.

Guidelines and limitations for Traffic Analytics troubleshooting rules

- When upgrading to NX-OS Release 10.5(1)F using a nondisruptive upgrade, the default value of **filter export-interval** is derived from the NetFlow **flow timeout** value.

Guidelines and limitations for Traffic Analytics interface filter and VRF filter

- The TA interface filter is not supported for loopback, tunnel interfaces (such as NVE), and management interfaces.
- The TA interface filter is not supported for L3 subinterfaces and L3 port-channel (PO) subinterfaces.
- The VRF filter is not supported for default and management VRFs.
- If TA interface filters and VRF filters are configured, TA interface filters take precedence.

Guidelines and limitations for ECN detection for Traffic Analytics

- Beginning with NX-OS Release 10.5(3)F, the ECN Detection for Traffic Analytics feature is supported on:
  - Nexus 9300-FX3, GX, GX2, H2R, and H1 platform switches.
  - Nexus 9700-FX and GX Line Cards.
  - Nexus 9500 EOR switches with GX and FX line cards
- This feature is designed for networks that require detailed congestion monitoring, especially for real-time applications. Configure Traffic Analytics to focus on detecting ECN bits.
- ECN detection is only supported on switches managed by Network Insights Resources (NIR).

Configure Traffic Analytics

Enable and configure the traffic analytics feature to monitor and analyze network flows.

You can configure the traffic analytics feature only in mixed mode.

Beginning with NX-OS Release 10.5(1)F, Traffic Analytics flows can be marked as troubleshoot flows for debugging purposes, and TA flows are exported to the Nexus Dashboard at a faster interval rate.

In the following example, the troubleshoot flows are defined in both IPv4 and IPv6 ACL lists and are attached to a flow filter. The flow filter has been enabled system-wide under the flow system configuration.

Before you begin

Ensure that you are in mixed mode before enabling the traffic analytics feature. To enable the mixed mode, use the following commands. For more information on mixed mode, see Mixed Mode, on page 475:

```
(Config)#feature netflow
(Config)#feature analytics
```

Procedure

Step 1 Configure traffic analytics feature with higher cadence support.

Example:

```
ip access-list ipv4-global_filter
  statistics per-entry
  1 permit ip 192.0.2.1/32 198.51.100.1/32
```

```
  2 permit ip 198.51.100.1/32 192.0.2.1/32
  3 permit ip 203.0.113.1/32 192.0.2.2/32
  4 permit ip 192.0.2.2/32 203.0.113.1/32

ipv6 access-list ipv6-global_filter
  statistics per-entry
  1 permit ipv6 2001:DB8::2/128 2001:DB8:1::2/128
  2 permit ipv6 2001:DB8:1::2/128 2001:DB8::2/128
  3 permit ipv6 2001:DB8:2::2/128 2001:DB8:3::2/128
  4 permit ipv6 2001:DB8:3::2/128 2001:DB8:2::2/128

flow filter global_filter
  ipv4 ipv4-global_filter
  ipv6 ipv6-global_filter


switch(config)# feature netflow
switch(config)# feature analytics
switch(config)# analytics
switch(config-analytics)#

switch(config-analytics)#  flow traffic-analytics
switch(config-analytics-traffic-analytics)#  db-size 200
switch(config-analytics-traffic-analytics)#  filter export-interval 30
switch(config-analytics-traffic-analytics)#  flow system config
switch(config-analytics-system)#  traffic-analytics
switch(config-analytics-system)#  monitor monitor input
switch(config-analytics-system)#  profile profile
switch(config-analytics-system)#  event event
switch(config-analytics-system)#  filter global_filter
```

Step 2 Use the **flow filter**<*filter*> command to configure traffic analytics for ingress interface.

Example:

```
switch(config)# interface Ethernet1/1
switch(config-if)# flow filter test
```

Step 3 Use the **flow filter**<*filter*>**output** command to configure traffic analytics for egress interface.

Note Before using egress filters, ensure that the **egress netflow tcam** region is carved.

Example:

```
switch(config)# interface Ethernet1/1
switch(config-if)# flow filter test output
```

Example for Traffic Analytics interface filter and VRF filter

This topic provides examples for configuring Traffic Analytics interface filters and VRF filters.

Interface filter configuration

This example shows how the interface filter configuration is performed.

```
ip access-list ipv4-l3_intf_filter
  statistics per-entry
  1 permit tcp 10.0.0.0/8 172.16.0.0/12 syn
  2 permit ip 10.0.0.0/8 172.16.0.0/12

ipv6 access-list ipv6-l3_intf_filter
  statistics per-entry
  1 permit tcp 2001:DB8::7/128 2001:DB8:1::7/128 syn
  2 permit ipv6 2001:DB8::7/128 2001:DB8:1::7/128

flow filter l3_filter
  ipv4 ipv4-l3_intf_filter
  ipv6 ipv6-l3_intf_filter

analytics

  flow traffic-analytics
    db-size 200
    filter export-interval 30
  flow system config
    traffic-analytics
    monitor monitor input
    profile profile
    event event

interface Ethernet1/63/1
  flow filter l3_filter
flow filter l3_filter output

switch(config-analytics)# show running-config inter e 1/63/1

interface Ethernet1/63/1
  vrf member vrf1
  flow filter l3_filter
  ip address 10.0.0.1/24
  ipv6 address 2001:DB8::1/64
  no shutdown
```

VRF filter configuration

This example shows how the VRF filter configuration is performed.

```
ip access-list ipv4-vrf1_filter
  statistics per-entry
  1 permit tcp 192.0.2.1/32 198.51.100.1/32 syn
  2 permit tcp 198.51.100.1/32 192.0.2.1/32 ack syn

ipv6 access-list ipv6-vrf1_filter
  statistics per-entry
  1 permit tcp 2001:DB8::9/128 2001:DB8::A/128 syn
  2 permit tcp 2001:DB8::A/128 2001:DB8::9/128 ack syn

flow filter vrf1_filter
  ipv4 ipv4-vrf1_filter
  ipv6 ipv6-vrf1_filter

analytics

  flow traffic-analytics
    db-size 200
    filter export-interval 30
  flow system config
    traffic-analytics
    monitor monitor input
```

```
    profile profile
    event event

vrf context vrf1

  flow filter vrf1_filter
flow filter vrf1_filter output
```

Example for Traffic Analytics

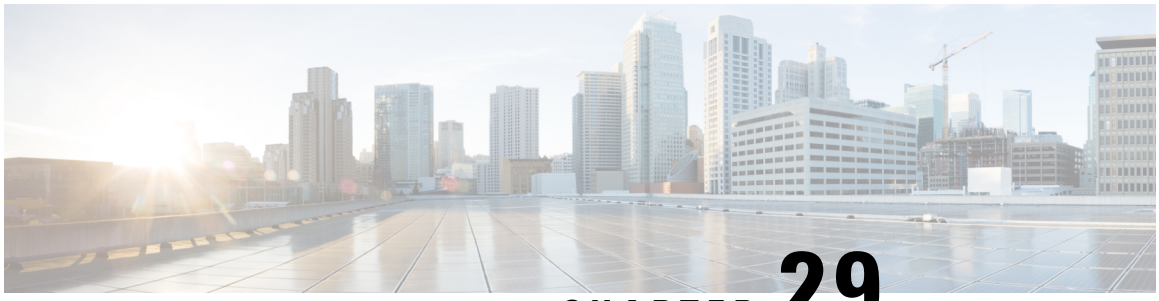
This section provides an example and explanation for configuring and troubleshooting the traffic analytics export interval.

This example displays the output of the troubleshoot flows export interval.

```
switch(config-analytics-traffic-analytics)# show flow traffic-analytics
Traffic Analytics:
    Service DB Size: 200
    Troubleshoot Export Interval: 30
```

The **filter export-interval** command allows setting the troubleshoot timer with a range of 10 to 60 seconds. The default value for this timer is set to 10 seconds.

The **no filter export-interval** resets the troubleshoot timer range to default value of 60 seconds.



CHAPTER 29

Sampled Flow

- Overview, on page 495
- Guidelines and Limitations, on page 496
- Enable sFlow Feature, on page 500
- Configure Sampling Rate, on page 502
- Configure Maximum Sampled Size, on page 502
- Configure Counter Poll Interval, on page 503
- Configure Maximum Datagram Size, on page 504
- Configure sFlow Collector Address, on page 505
- Configure Collector Port, on page 506
- Configure Agent Address, on page 507
- Configure Sampling Data Source, on page 508
- Configure Extended BGP (Gateway), on page 509
- Related Documents, on page 509

Overview

Sampled flow (sFlow) is a sampled flow monitoring technology that enables you to monitor real-time traffic in data networks with switches and routers, employs the sampling mechanism in sFlow agent software to monitor traffic, and forwards sample data to a central data collector.

- Enables monitoring of real-time traffic in data networks with switches and routers
- Employs a sampling mechanism in sFlow agent software to monitor traffic
- Forwards sample data to a central data collector

For more information about sFlow, see RFC 3176 .

sFlow Agent

The sFlow agent, which is embedded in the Cisco NX-OS software, periodically samples or polls the interface counters that are associated with a data source of the sampled packets. The data source can be an Ethernet interface, an EtherChannel interface, or a range of Ethernet interfaces. The sFlow agent queries the Ethernet port manager for the respective EtherChannel membership information and also receives notifications from the Ethernet port manager for membership changes.

When you enable sFlow sampling, based on the sampling rate and the hardware internal random number, the ingress packets and egress packets are sent to the CPU as an sFlow-sampled packet. The sFlow agent processes the sampled packets and sends an sFlow datagram to the sFlow analyzer. In addition to the original sampled packet, an sFlow datagram includes information about the ingress port, the egress port, and the original packet length. An sFlow datagram can have multiple sFlow samples.

Default Settings

The following table lists the default settings for sFlow parameters.

Table 36: Default sFlow Parameters

Parameters	Default
sFlow sampling rate	4096
sFlow sampling size	128
sFlow counter poll interval	20
sFlow maximum datagram size	1400
sFlow collector IP address	0.0.0.0
sFlow collector port	6343
sFlow agent IP address	0.0.0.0

Guidelines and Limitations

Prerequisites

- For Cisco Nexus 9332PQ, 9372PX, 9372TX, and 93120TX switches and for Cisco Nexus 9396PX, 9396TX, and 93128TX switches with the N9K-M6PQ generic expansion module (GEM), you must configure the sFlow and SPAN ACL TCAM region sizes for any uplink ports that are to be configured as an sFlow data source. To do so, use the **hardware access-list tcam region sflow** and **hardware access-list tcam region span** commands. See Configuring ACL TCAM Region Sizes for more information.

Note By default, the sflow region size is zero, and the span region size is non-zero. You need to configure the sflow region to 256 and allocate enough entries to the span region in order to configure the port as an sFlow data source.

- Egress sFlow of multicast traffic requires **hardware multicast global-tx-span** configuration.

General Guidelines



Note For scale information, see the release-specific *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide* .

- When sFlow is used to sample the Rx traffic from FEX HIF ports, additional VNTAG and 802.1q tags are present in the sampled traffic.
-
- sFlow is a software-driven feature, hardware only sends copies of traffic from the sFlow source interfaces to the CPU for further processing. Elevated CPU usage is expected. sFlow traffic sent to the CPU by hardware is rate-limited to protect the CPU.
- When you enable sFlow for an interface, it is enabled for both ingress and egress. You cannot enable sFlow for only ingress or only egress.

  For Cisco Nexus 9508 switches with Cisco Nexus 9636C-R and 9636Q-R line cards, sFlow can be enabled for an interface only in the ingress direction.
- The storm control feature does not work if you enable storm control on an interface where sFlow is also enabled.
- sFlow counters increment even for control packets that ingress on the sFlow data-source interfaces. These packets may be sampled and send out as sFlow datagrams (similar to data plane traffic).
- NX-OS provides flexible forwarding templates to utilize the hardware resources according to customer needs. For sFlow ingress IPv6 sampling to fill BGP information correctly in the sFlow record, a template which has all IPv6 routes on the line-card has to be selected. For example, customers can configure **system routing template-mpls-heavy** . For more information, please refer to the Cisco Nexus 9000 Series NX-OS Command Reference (Configuration Commands), Release 9.3(x). For command to take effect, system needs to be rebooted. This is applicable on GX modular chassis.

Configuration Guidelines

1. If at least one sFlow data source is configured, the SPAN sessions cannot be brought up.
   - If at least one SPAN session is configured as **no shut** , sFlow data sources cannot be added.
   - The sampling mode that is used for sFlow is based on an algorithm that is known as LFSR. Due to the use of LFSR, it is not guaranteed that one in every few packets are sampled with the sampling rate of n. However, the number of packets that are sampled is equal to the total packets over a period of time.
2. When sFlow and SPAN are configured on the same interface, and the hardware rate-limiter is configured for sFlow, the Rate-Limiter Drops counter in the output of the **show hardware rate-limiter** command displays more drops than expected.
3. We recommend you configure the sampling rate that is based on the sFlow configuration and traffic in the system.
4. When sFlow is configured on N9K-C9508-FM-G with the N9K-X9716D-GX line card, disable sFlow before configuring SPAN sessions.

5. When ECMP is configured in BGP and in case of ECMP destination routes, the next-hop information in the extended gateway record of the exported sFlow record will be 0. Other BGP information like Autonomous System will be derived from the first path. The output interface in the sFlow record will be set to 0 (unknown) to indicate that the flow could be through any of the paths.
6. When you configure SPAN Tx on the same interface as the sFlow data-source port, the effective sFlow process samples are halved for that source port from the configured sampling rate.

Supported Features

1. The switch supports only one sFlow collector.
2. sFlow supports sampling IPv6 traffic .
3. The following Cisco Nexus switches support sFlow and SPAN together:
   - N9336C-FX2
   - N93240YC-FX2
   - N93360YC-FX2
4. Nexus 9000-EX, FX, GX family of switches only support sampling at the following values: 4096, 8192, 16384, 32768, 65536. Configuring values other than these results in the value being rounded off to the next supported value.

Not Supported Features

- sFlow is not supported on the SVIs.
- Subinterfaces are not supported for sFlow.
- sFlow does not support egress sampling for multicast, broadcast, or unknown unicast packets.
- sFlow and Network Address Translation (NAT) are not supported on the same port.

Limitations for Cisco Nexus 9804 switches

- For egress sampled packet, re-written information is not available in sFlow record
- Egress sFlow is not supported for directly connected host
- sFlow is not supported on the sub-interface traffic

Release Specification Limitations

Beginning with Cisco NX-OS Release 9.3(3)

- Cisco Nexus 9300-GX platform switches support both sFlow and SPAN together.

Beginning with Cisco NX-OS Release 10.1(1)

- sFlow and SPAN are supported on Cisco N9K-C93180YC-FX3 platform switches.

Beginning with Cisco NX-OS Release 10.1(2)

- sFlow is supported on the Cisco N9K-X9624D-R2 line card.
- sFlow supports VXLAN traffic on the Cisco N9K-C9508-FM-G cloud-scale fabric module with the N9K-X9716D-GX line card.

Beginning with Cisco NX-OS Release 10.2(1)

- sFlow Extended BGP (Gateway) is supported on the Cisco N9K-C93600CD-GX, N9K-C93240YC-FX2, N9K-C93180YC-EX, N9K-C93180YC-FX, N9K-C93180YC-FX3S, N9K-93600CD-GX platform switches, and Nexus 9500 with N9K-X9716D-GX and N9K-X9736C-FX line cards.
- Extended BGP data can now be collected. In order for sFlow to collect this data, a non-SVI Layer 3 interface such as a physical interface or port-channel must be configured as the sFlow source.
- Beginning with Cisco NX-OS Release 10.2(1q)F, sFlow is supported on the Cisco N9K-C9332D-GX2B platform switches.

Beginning with Cisco NX-OS Release 10.2(3)F

- sFlow flow-cache size is increased from 3k route entries in earlier releases to 30k v4 and 30k v6 route entries. This feature is supported on Cisco Nexus C93600CD-GX, C93240YC-FX2, C93180YC-EX, C93180YC-FX, C93180YC-FX3S, 93600CD-GX, and X9716D-GX platform switches.

Beginning with Cisco NX-OS Release 10.3(1)F

- sFlow is supported on the Cisco Nexus 9808 platform switches.
- For egress sampled packet, re-written information is not available in sFlow record.
- Egress sFlow is not supported for directly connected host. However, beginning with Cisco NX-OS Release 10.4(2)F, Egress sFlow is supported for directly connected host.
- sFlow is not supported on the sub-interface traffic.
- sFlow supports IPv6 collector. However, at a time, only one collector can be configured, either IPv4 or IPv6. Also, the source ip address and the collector ip address must belong to the same address family, that is, either IPv4 or IPv6 address family.

Beginning with Cisco NX-OS Release 10.4(1)F

- sFlow is supported on Cisco Nexus 9804 switch, Cisco Nexus C9332D-H2R switch and Cisco Nexus X98900CD-A and X9836DM-A line cards with Cisco Nexus 9808 and 9804 switches.

Beginning with Cisco NX-OS Release 10.4(2)F

- sFlow is supported on Cisco Nexus 93400LD-H1 platform switches.

Beginning with Cisco NX-OS Release 10.5(3)F

- sFlow is supported on Cisco N9364E-SG2-O and N9364E-SG2-Q TOR switches. However, these switches do not support sFlow on the sub-interface traffic. For egress sampled packet, re-written information is not available in the sFlow record.

Beginning with Cisco NX-OS Release 10.6(1)F

- sFlow is supported on the Cisco N9336C-SE1 switch.
- For egress sampled packet, re-written information is not available in sFlow record.

- sFlow is not supported on the sub-interface traffic.
- Egress Layer 2 source interface is not supported.

N9300 Smart Switch Guidelines

Beginning with NX-OS Release 10.6(2)F, sFlow is supported on the N9324C-SE1U and N9348Y2C6D-SE1U switches. The guidelines and limitations include:

- For egress sampled packet, re-written information is not available in sFlow record.
- Egress Layer 2 source interface is not supported.
- sFlow is not supported on the sub-interface traffic.
- sFLow is not supported on Service-Ethernet or Service-Port-channels.

Enable sFlow Feature

You must enable the sFlow feature before you can configure sFlow settings on the switch.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Enable or disable sFlow using the command [**no**] **feature sflow**

Example:

```
switch(config)# feature sflow
```

Step 3 Display the enabled and disabled features using the command **show feature**

Example:

```
switch(config)# show feature
```

This is an optional step.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

This example shows how to configure sFlow:

```
feature sflow
sflow sampling-rate 4096
sflow max-sampled-size 200
sflow counter-poll-interval 100
sflow max-datagram-size 2000
sflow collector-port 7000
sflow agent-ip 192.0.2.3
sflow collector-ip 192.0.2.5 vrf management
sflow data-source interface ethernet 1/5
```

What to do next

Use these commands to display the sFlow configuration.

Table 37: sFlow Show Commands

Command	Purpose
show sflow	Displays all the data sources of the sFlow samplers and the sFlow agent configuration.
show process	Verifies whether the sFlow process is running.
show system internal sflow consistency issu-pss	Verifies that there is no discrepancy before and after an upgrade. **Note** Post ISSU from an earlier release to NX-OS Release 10.5(3)F or later releases, this command requires that the `sflow_pss_cc_upgrade_post_snapshot.txt` and `sflow_pss_cc_upgrade_pre_snapshot.txt` files to be present. If either of the files are missing, the command does not display the expected output, instead displays information about the missing files, either pre upgrade, post upgrade, or both the snapshot files.
show running-config sflow [**all**]	Displays the current sFlow running configuration.

Use the **show sflow statistics** command to display the sFlow statistics. Use the following commands to clear the sFlow statistics:

Command	Description
clear sflow statistics	Clears most of the sFlow statistics from the **show sflow statistics** command.
clear counters interface all	Clears the Total Packets field from the **show sflow statistics** command.
clear hardware rate-limiter sflow	Clears the Total Samples field from the **show sflow statistics** command.

Configure Sampling Rate

Configure the sampling rate for sFlow.

You can configure the sampling rate for sFlow.

Before you begin

Make sure that you have enabled sFlow.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
  switch(config)#
```

Step 2 Configure the sFlow sampling rate for packets using the command [**no**] **sflow sampling-rate** *sampling-rate*

Example:

```
switch(config)# sflow sampling-rate 50000
```

The *sampling-rate* can be an integer between 4096 and 1000000000.

Note Nexus 9000-EX, FX, GX, FX2, FX3, GX2 , and HX family of switches support sampling only at the following values: 4096, 8192, 16384, 32768, 65536. Configuring values other than these will result in the value being rounded off to the next supported value.

Step 3 Display the sFlow configuration using the command **show sflow**

Example:

```
switch(config)# show sflow
```

This is an optional step.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure Maximum Sampled Size

You can configure the maximum number of bytes that should be copied from a sampled packet.

Before you begin

Make sure that you have enabled sFlow.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure the sFlow maximum sampling size using the command [**no**] **sflow max-sampled-size** *sampling-size*

Example:

```
switch(config)# sflow max-sampled-size 200
```

The range for the *sampling-size* is from 64 to 256 bytes.

Step 3 Display the sFlow configuration using the command **show sflow**

Example:

```
switch(config)# show sflow
```

This is an optional step.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure Counter Poll Interval

Before you begin

Make sure that you have enabled sFlow.

You can configure the maximum number of seconds between successive samples of the counters that are associated with the data source. A sampling interval of 0 disables counter sampling.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure the sFlow poll interval for an interface using the command [**no**] **sflow counter-poll-interval** *poll-interval*

Example:

```
switch(config)# sflow counter-poll-interval 100
```

The range for *poll-interval* is from 0 to 2147483647 seconds.

Step 3 Display the sFlow configuration using the command **show sflow**

Example:

```
switch(config)# show sflow
```

This is an optional step.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure Maximum Datagram Size

Configure the maximum number of data bytes that can be sent in a single sample datagram.

You can configure the maximum number of data bytes that can be sent in a single sample datagram.

Before you begin

Make sure that you have enabled sFlow.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure the sFlow maximum datagram size using the command [**no**] **sflow max-datagram-size** *datagram-size*

Example:

```
switch(config)# sflow max-datagram-size 2000
```

The range for the *datagram-size* is from 200 to 9000 bytes.

Step 3 Display the sFlow configuration using the command **show sflow**

Example:

```
switch(config)# show sflow
```

This is an optional step.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure sFlow Collector Address

Configure the IPv4 or IPv6 address of the sFlow data collector that is connected to the management port.

You can configure the IPv4 or IPv6 address of the sFlow data collector that is connected to the management port.

Before you begin

Make sure that you have enabled sFlow.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure the IPv4 or IPv6 address for the sFlow collector using the command [**no**] **sflow collector-ip** *ip-address* **vrf** *vrf* [**source** *ip-address*]

Example:

```
switch(config)# sflow collector-ip 192.0.2.5 vrf management
switch(config)# sflow collector-ip 2001::1 vrf management
```

If the IP address is set to 0.0.0.0, all samples will be dropped. The *vrf* can be one of the following:

- A user-defined VRF name—You can specify a maximum of 32 alphanumeric characters.
- **vrf management** —You must use this option if the sFlow data collector is on the network connected to the management port.
- **vrf default** —You must use this option if the sFlow data collector is on the network connected to the front-panel ports.

The **source** *ip-address* option causes the sent sFlow datagram to use the source IP address as the IP packet source address. The source IP address has to be already configured on one of the switch local interfaces; otherwise, an error

message appears. If the interface with the source IP address is changed or removed after this option is configured, the sFlow datagram will no longer be sent out, and an event history error and syslog error will be logged. When the **source** *ip-address* option is not configured, Cisco NX-OS picks the IP packet source address automatically for the sent sFlow datagram.

Step 3 Display the sFlow configuration using the command **show sflow**

Example:

```
switch(config)# show sflow
```

This is an optional step.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure Collector Port

You can configure the destination port for sFlow datagrams.

Before you begin

Make sure that you have enabled sFlow.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure the UDP port of the sFlow collector using the command [**no**] **sflow collector-port** *collector-port*

Example:

```
switch(config)# sflow collector-port 7000
```

The range for the *collector-port* is from 1 to 65535.

Step 3 Display the sFlow configuration using the command **show sflow**

Example:

```
switch(config)# show sflow
```

This is an optional step.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure Agent Address

Configure the sFlow agent address to enable sFlow functionality on the device.

You can configure the IPv4 or IPv6 address of the sFlow agent.

Before you begin

Make sure that you have enabled sFlow.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure the IPv4 or IPv6 address of the sFlow agent using the command [**no**] **sflow agent-ip** *ip-address*

Example:

```
switch(config)# sflow agent-ip 192.0.2.3
```

```
switch(config)# sflow agent-ip 2001::10
```

The default IP address is 0.0.0.0, which means that all samples will be dropped. You must specify a valid IP address to enable sFlow functionality.

Note This IP address is not necessarily the source IP address for sending the sFlow datagram to the collector.

The agent ip address and the collector ip address must belong to the same address family, that is, either IPv4 or IPv6 address family.

Step 3 Display the sFlow configuration using the command **show sflow**

Example:

```
switch(config)# show sflow
```

This is an optional step.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure Sampling Data Source

Before you begin

Make sure that you have enabled sFlow. If you want to use a port channel as the data source, make sure that you have already configured the port channel and you know the port channel number. Make sure that the sFlow and SPAN ACL TCAM region sizes are configured for any uplink ports that are to be configured as an sFlow data source on the following devices: Cisco Nexus 9332PQ, 9372PX, 9372TX, and 93120TX switches and Cisco Nexus 9396PX, 9396TX, and 93128TX switches with the N9K-M6PQ generic expansion module (GEM).

You can configure the source of the data for the sFlow sampler as an Ethernet port, a range of Ethernet ports, or a port channel.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure the sFlow sampling data source using the command [**no**] **sflow data-source interface** [**ethernet** *slot* / *port* [- *port*] | **port-channel** *channel-number*]

Example:

```
switch(config)# sflow data-source interface ethernet 1/5-12
```

For an Ethernet data source, *slot* is the slot number, and *port* can be either a single port number or a range of ports designated as *port-port* .

Step 3 Display the sFlow configuration using the command **show sflow**

Example:

```
switch(config)# show sflow
```

This is an optional step.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure Extended BGP (Gateway)

You can configure sFlow Extended BGP on the switch.

Before you begin

Make sure that you have enabled sFlow.

Make sure that the source port is a non-SVI Layer 3 interface, such as a physical interface or port-channel.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Configure extended bgp on the switch using the command [**no**] **sflow extended bgp**

Example:

```
switch(config)# sflow extended bgp
```

Sampled sFlow packets with destination IP address to BGP installed routes will include extended gateway (bgp) data in the exported sFlow record.

Step 3 Display the sFlow configuration using the command **show sflow**

Example:

```
switch(config)# show sflow
```

This is an optional step.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

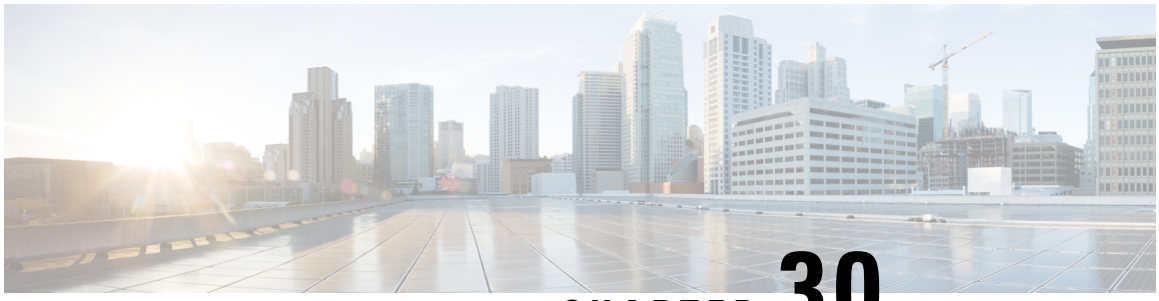
```
switch(config)# copy running-config startup-config
```

This is an optional step.

Related Documents

This section provides a reference to related documents for ACL TCAM regions.

Related Topic	Document Title
ACL TCAM regions	Configuring IP ACLs



CHAPTER 30

Test Access Points Aggregation

- About TAP Aggregation, on page 511
- Enable TAP Aggregation for Line Cards, on page 514
- Configure TAP Aggregation Policy, on page 515
- Attach TAP Aggregation Policy to Interface, on page 516
- Configure Selective Q-in-Q with Provider VLANs, on page 517
- About MPLS Stripping, on page 520
- Enable MPLS Stripping, on page 521
- Configure Incoming Port for VLAN Tag, on page 522
- Add and Delete MPLS Labels, on page 523
- Configure Destination MAC Address, on page 524
- Configure MPLS Label Aging, on page 525
- MPLS Stripping Configuration Commands and Examples , on page 526
- Related Documents, on page 528

About TAP Aggregation

Test access points (TAP) aggregation is an alternative solution to help with monitoring and troubleshooting tasks in the data center. It works by designating a device to allow the aggregation of multiple test access points and to connect to multiple monitoring systems. TAP aggregation switches link all of the monitoring devices to specific points in the network fabric that handle the packets that need to be observed.

In the TAP aggregation switch solution, a Cisco Nexus 9000 Series switch is connected to various points in the network at which packet monitoring is advantageous. From each network element, you can use switched port analyzer (SPAN) ports or optical TAPs to send traffic flows directly to this TAP aggregation switch. The TAP aggregation switch is directly connected to all of the analysis tools used to monitor the events in the network fabric. These monitoring devices include remote monitor (RMON) probes, application firewalls, IPS devices, and packet sniffer tools.

You can configure the TAP aggregation switch to filter specific traffic and redirect it to one or more tools. In order to redirect the traffic to multiple interfaces, a multicast group is created internally on the switch, and the interfaces that are part of the redirect list are added as member ports. When an access control list (ACL) policy with the redirect action is applied to an interface, the traffic matching the ACL rule is redirected to the internal multicast group that is created.

Network TAPs

You can use various methods to monitor packets. One method uses physical hardware test access points (TAPs).

Network TAPs can be extremely useful in monitoring traffic because they provide direct inline access to data that flows through the network. In many cases, a third party monitors the traffic between two points in the network. If the network between points A and B consists of a physical cable, a network TAP might be the best way to accomplish this monitoring. The network TAP has at least three ports: an A port, a B port, and a monitor port. A TAP inserted between the A and B ports passes all traffic through unimpeded, but it also copies that same data to its monitor port, which could enable a third party to listen.

TAPs have the following benefits:

- They can handle full-duplex data transmission.
- They are unobtrusive and not detectable by the network (with no physical or logical addressing).
- Some TAPs support full inline power with the capability to build a distributed TAP.

If you are trying to gain visibility into the server-to-server data communication at the edge or virtual edge of your network or to provide a copy of traffic to the Intrusion Prevention System (IPS) appliance at the Internet edge of your network, you can use network TAPs nearly anywhere in the environment. However, this deployment can add significant costs, operation complexities, and cabling challenges in a large-scale environment.

Guidelines and Limitations

General Guidelines

- TAP aggregation filters on MPLS tags are not supported on the following Cisco Nexus Series switches, line cards, and fabric modules:

Table 38: Cisco Nexus 9000 Series Switches

Cisco Nexus 3164Q-40GE	Cisco Nexus 9372PX	Cisco Nexus 9372PX-E
Cisco Nexus 9372TX	Cisco Nexus 9372TX-E	Cisco Nexus 9332PQ
Cisco Nexus 3232C	Cisco Nexus 93120TX	Cisco Nexus 31128PQ
Cisco Nexus 3264Q-S	—	—

Table 39: Cisco Nexus 9000 Series Line Cards and Fabric Modules

N9K-M6PQ	N9K-X9632PC-QSFP100	N9K-X9536PQ
N9K-X9432C-S	N9K-C93128TX	N9K-C9396PX
N9K-X9432PQ	N9K-X9464TX	—

- Only Layer 2 interfaces support the TAP aggregation policy. You can apply the policy to a Layer 3 interface, but the policy becomes nonfunctional.
- The redirect port must be part of the same VLAN as the source (TAP) port.

- Each rule must be associated with only one unique match criterion.
- When you enter a list of interfaces for the TAP aggregation policy, you must separate them with commas but no spaces. For example, port-channel50, ethernet1/12, port-channel20.
- When you specify target interfaces in a policy, make sure that you enter the whole interface type and not an abbreviated version. For example, make sure that you enter **ethernet1/1** instead of **eth1/1** and **port-channel50** instead of **po50**.
- HTTP requests with **tcp-option-length** and **VLAN ID** filters simultaneously are not supported. Traffic match against ACE may not work if you configure both filters at a time.
- When configuring ACL entries with redirect to port-channels that are yet to be configured, the user must take care to configure the specified port-channels at a later point of time.
- To allow double VLAN tags on ingress interface, the **switchport trunk allow-multi-tag** command must be configured correctly as mentioned below:
  - On Cisco Nexus 9300-FX2 switches, this command must be used only if NDB is configured.
  - On Cisco Nexus 9300-GX/GX2 switches, this command is not required if NDB is configured.

Supported Features



Note For scale information, see the release-specific *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*.

- Supported on all Cisco Nexus 9000 Series switches and the 3164Q, 31128PQ, 3232C, and 3264Q switches.
- Supported on 100G ports.
- Supports only on switch ports and only in the ingress direction.
- Supported on Cisco Nexus 9300-FX, 9300-FX2, 9300-FX3, 9300-GX, 9300-GX2, 9300-HX, 9500-EX, and 9500-FX platform switches.
- Maximum redirect ports supported are 32 interfaces.

Release Specific Guidelines

- Beginning with Cisco NX-OS Release 9.2(1), TAP aggregation filters on MPLS tags are supported on the following Cisco Nexus platform switches:
  - Cisco Nexus 9200 platform switches.
  - Cisco Nexus 9300 platform switches.
  - Cisco Nexus 9500 switches.
- Beginning with Cisco NX-OS Release 10.2(1)F, the TAP aggregation feature is licensed and requires you to configure feature tap-aggregation before configuring related CLIs. However, this feature is auto-generated during ISSU infra-convert phase of sysmgr if any tap-aggregation dependent CLI usage is found in the earlier configurations. This feature is supported on all Cisco Nexus 9000 Series switches. For more information about licensing, refer to **Cisco Nexus 9000 NX-OS Smart Licensing Using Policy Guide.**

- Beginning with Cisco NX-OS Release 10.2(2)F, ensure that you configure the **mode tap-aggregation** command before attaching TapAgg ACLs on L2 interface.
- Beginning with Cisco NX-OS Release 10.3(1)F, on the interface with selective Q-in-Q trunk mode the provider VLAN tagging is supported on the Cisco Nexus 9300-GX, N9K-C9504-FM-G, and N9K-C9508-FM-G switches and N9K-X9716D-GX line cards with below limitations:
  - If VXLAN is enabled, this feature is not supported.
  - You can have maximum 7000 outer VXLAN late entries at overall system level and 4000 entries per port.

Limitations for hardware acl tap-aggCommand

- This command is required on Cisco Nexus 9300-GX platform switches for the header stripping functionality to work. After the configuration a switch reload is required.
- If **mode tap-aggregation** is configured under interfaces on Cisco Nexus 9300-GX, -H2R, and -H1 switches, this command is mandatory.
- This command is not required on Cisco Nexus 9300-FX3 platforms.
- Beginning with NX-OS Release 10.4(3), this command is required on Cisco Nexus 9300-FX3 platforms before you enable **mode tap-aggregation**

Enable TAP Aggregation for Line Cards

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
                        switch(config)#
```

Step 2 Enable TAP aggregation for Cisco Nexus 9700-EX and 9700-FX line cards using the command [**no**] **hardware acl tap-agg**

Example:

```
switch(config)# hardware acl tap-agg
```

This command is also needed on Cisco Nexus 9300-GX , 9300-HX, 9300-FX3, and 9300-GX2 platform switches and may require reload.

Step 3 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure TAP Aggregation Policy

Use this task to configure a TAP aggregation policy on an IP or MAC ACL, allowing you to redirect and monitor network traffic as required.

You can configure a TAP aggregation policy on an IP access control list (ACL) or on a MAC ACL.

Before you begin

You must configure the ACL TCAM region size for IPv4 port ACLs or MAC port ACLs using the **hardware access-list tcam region** { *ifacl* | *mac-ifacl* } command. Configure the ACL TCAM region size for IPv6 port ACLs using the command, **hardware access-list team region ipv6-ifcal** .

For information, see the "Configuring ACL TCAM Region Sizes" in the Cisco Nexus 9000 Series NX-OS Security Configuration Guide .

Note By default the region size for both ifacl and mac-ifacl is zero. You need to allocate enough entries to the ifacl or mac-ifacl region to support TAP aggregation.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Configure CLIs related to tap-aggregation using the command **feature tap-aggregation**

Example:

```
switch(config)# feature tap-aggregation
 switch(config)#
```

Note Beginning with Cisco NX-OS Release 10.2(1)F, for software upgrades from earlier releases to the newer NX-OS release with this feature, if ISSU is completed on a supported matrix, the feature tap-aggregation configuration is automatically generated.

Step 3 Enter one of the following commands:

- **ip access-list** *access-list-name*
- **mac access-list** *access-list-name*

Example:

```
switch(config)# ip access-list test
 switch(config-acl)#
 switch(config)# mac access-list mactap1
 switch(config-mac-acl)#
```

Creates an IPACL and enters IP access list configuration mode or creates a MAC ACL and enters MAC access list configuration mode.

Step 4 Start recording statistics for how many packets are permitted or denied by each entry using the command **statistics per-entry**

Example:

```
switch(config-acl)# statistics per-entry
```

This is an optional step.

Step 5 Create an IP or MAC ACL rule that permits traffic to be redirected per its conditions using the command [**no**] **permit** *protocol source destination* **redirect** *interfaces*

Example:

```
switch(config-acl)# permit ip any any redirect ethernet1/8
```

The **no** version of this command removes the permit rule from the policy.



Note When you enter an interface for the TAP aggregation policy, do not abbreviate it. When you enter a list of interfaces, separate them with commas but no spaces.

Step 6 Display all IPv4 or MAC ACLs or a specific IPv4 or MAC ACL using one of the following commands:

- **show ip access-lists** [*access-list-name*]
- **show mac access-lists** [*access-list-name*]

Example:

```
switch(config-acl)# show ip access-lists test
 switch(config-mac-acl)# show mac access-lists mactap1
```

This is an optional step.

Step 7 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config-acl)# copy running-config startup-config
```

This is an optional step.

Attach TAP Aggregation Policy to Interface

Use this task to configure TAP aggregation policies on IPv4 and MAC ACLs, and to apply or attach these policies to interfaces as required for traffic monitoring and redirection.

Procedure

Step 1 Configure a TAP aggregation policy on an IPv4 ACL.

Example:

```
switch# configure terminal
 switch(config)# feature tap-aggregation
 switch(config)# ip access-list test
 switch(config-acl)# 10 deny ip 100.1.1/24 any
 switch(config-acl)# 20 permit tcp any eq www any redirect port-channel4
 switch(config-acl)# 30 permit ip any any redirect
Ethernet1/1,Ethernet1/2,port-channel7,port-channel8,Ethernet1/12,Ethernet1/13
 switch(config-acl)# show ip access-lists test
 IP access list test
 10 deny ip 100.1.1/24 any
 20 permit tcp any eq www any redirect port-channel4
       30 permit ip any any redirect
Ethernet1/1,Ethernet1/2,port-channel7,port-channel8,Ethernet1/12,Ethernet1/13
```

Step 2 Apply a TAP aggregation policy to an IPv4 ACL with UDF-based match.

Example:

```
switch# configure terminal
 switch(config)# feature tap-aggregation
 switch(config)# ip access-list tap_agg
 switch(config-acl)# statistics per-entry
 switch(config-acl)# 10 permit ip any any redirect Ethernet1/1
 switch(config-acl)# 20 deny ip any any
 switch# configure terminal
 switch(config)# interface Ethernet1/4
 switch(config-if)# ip port access-group tap_agg in
 switch(config-if)# switchport
 switch(config-if)# no shutdown
```

Step 3 Configure a TAP aggregation policy on a MAC ACL.

Example:

```
switch# configure terminal
 switch(config)# feature tap-aggregation
 switch(config)# mac access-list mactap1
 switch(config-mac-acl)# 10 permit any any 0x86dd redirect port-channel1
       switch(config-mac-acl)# show mac access-lists mactap1
 MAC access list mactap1
 10 permit any any 0x86dd redirect port-channel1
```

Step 4 Attach a TAP aggregation policy to a Layer 2 interface.

Example:

```
switch# configure terminal
 switch(config)# interface ethernet 1/2
 switch(config-if)# ip port access-group test in
 switch(config-if)#
```

Configure Selective Q-in-Q with Provider VLANs

Before you begin

You must configure provider VLANs

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
```

Step 2 Enter interface configuration mode for the interface connected to the service provider network using the command **interface** *interface-id*

Example:

```
switch(config)#
interface Ethernet1/1
```

You can enter a physical interface or an EtherChannel port channel.

Step 3 Set the interface as a Layer 2 switching port using the command **switchport** and set the interface as a Layer 2 trunk port using the command **switchport mode trunk**

Example:

```
switch(config if)#
switchport
```

Example:

```
switch(config-if)#
switchport mode trunk
```

Step 4 Enter one of the following commands:

- **switchport vlan mapping** *vlan-id-range* **dot1q-tunnel** *outer vlan-id*
- **switchport vlan mapping all dot1q-tunnel** *outer vlan-id*

Example:

```
switch(config-if)#
switchport vlan mapping all dot1q-tunnel 300
```

Enter the VLAN IDs to be mapped:

- vlan-id-range—The customer VLAN ID range(C-VLAN) entering the switch from the customer network. The range is from 1 to 4094. You can enter a string of VLAN-IDs.
- outer vlan-id—Enter the outer VLAN ID (S-VLAN) of the service provider network. The range is from 1 to 4094.

Step 5 Set the allowed VLANs for the trunk interface using the command **switchport trunk allowed vlan** *vlan_list*

Example:

```
switch(config-if)# switchport trunk allowed vlan 300
```

Step 6 Enter one of the following commands:

- [**no**] **ip port access-group** *access-list-name* **in**
- [**no**] **mac port access-group** *access-list-name* **in**

Example:

```
switch(config-if)# ip port access-group test in
                      switch(config-if)# mac port access-group test in
```

Applies an IPv4 or MAC ACL configured with TAP aggregation to the interface. The **no** form of this command removes the ACL from the interface.

Step 7 Allow the attachment of an ACL with the tap aggregation policy to the interface using the command **mode tap-aggregation**

Example:

```
switch(config-if)# mode tap-aggregation
                      switch(config-if)# no shutdown
```

This is an optional step.

Step 8 Copy the running configuration to the startup configuration using the command **copy running-config startup-config** and verify the mapping configuration using the command switch(config-if)# **show interfaces** *interface-id* **vlan mapping**

Example:

```
switch(config-if)# copy running-config startup-config
```

This is an optional step.

Step 9 Exit the configuration mode using the command switch(config-if)# **exit**

This example shows how to configure a TAP aggregation policy on an IPv4 ACL:

```
switch# configure terminal
switch(config)# feature tap-aggregation
switch(config)# ip access-list test
switch(config-acl)# 10 deny ip 100.1.1/24 any
switch(config-acl)# 20 permit tcp any eq www any redirect port-channel4
switch(config-acl)# 30 permit ip any any redirect
Ethernet1/1,Ethernet1/2,port-channel7,port-channel8,Ethernet1/12,Ethernet1/13
switch(config-acl)# show ip access-lists test
IP access list test
        10 deny ip 100.1.1/24 any
        20 permit tcp any eq www any redirect port-channel4
        30 permit ip any any redirect
Ethernet1/1,Ethernet1/2,port-channel7,port-channel8,Ethernet1/12,Ethernet1/13
```

This example shows how to configure a TAP aggregation policy on a MAC ACL:

```
switch# configure terminal
switch(config)# feature tap-aggregation
switch(config)# mac access-list mactap1
switch(config-mac-acl)# 10 permit any any 0x86dd redirect port-channel1
switch(config-mac-acl)# show mac access-lists mactap1
MAC access list mactap1
        10 permit any any 0x86dd redirect port-channel1
```

This example shows how to attach a TAP aggregation policy to a Layer 2 interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# ip port access-group test in
switch(config-if)#
```

What to do next

To display the TAP aggregation configuration information, perform one of the following tasks.

Command	Purpose
show ip access-lists [*access-list-name*]	Displays all IPv4 ACLs or a specific IPv4 ACL.
show mac access-lists [*access-list-name*]	Displays all MAC ACLs or a specific MAC ACL.

About MPLS Stripping

The ingress ports of Cisco Nexus 9000 Series switches receive various Multiprotocol Label Switching (MPLS) packet types. Each data packet in an MPLS network has one or more label headers. These packets are redirected on the basis of a redirect access control list (ACL).

A label is a short, four-byte, fixed-length, locally significant identifier that is used to identify a Forwarding Equivalence Class (FEC). The label that is put on a particular packet represents the FEC to which that packet is assigned. It has the following components:

- Label—Label value (unstructured), 20 bits
- Exp—Experimental use, 3 bits; currently used as a class of service (CoS) field
- S—Bottom of stack, 1 bit
- TTL—Time to live, 8 bits

Standard network monitoring devices cannot monitor and analyze the MPLS traffic. You need to enable the MPLS strip feature to allow the standard network monitoring tools to monitor the MPLS traffic. This feature strips off the MPLS label headers of the traffic and redirects the traffic to the monitoring devices.

Guidelines and Limitations

Note For scale information, see the release-specific *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*.

- Beginning from Cisco NX-OS Release 10.2(1)F, **feature tap-aggregation** must be enabled for all Tap Aggregation and stripping functions.
- Beginning with Cisco NX-OS Release 10.2(3)F, OFM-based MPLS stripping is added. The new OFM-based MPLS stripping and legacy implementation cannot co-exist. For more information, see the OFM-based MPLS header strip section under Header Stripping for Nexus Data Broker, on page 545.
- Beginning with Cisco NX-OS Release 10.3(2)F, EoMPLS label stripping is also supported on Cisco Nexus 9300-FX ToR switches.

-
- MPLS stripping is supported only for IPv4 traffic.
- Port-channel load balancing is supported for MPLS stripped packets.
- Layer 3 header-based hashing and Layer 4 header-based hashing are supported, but Layer 2 header-based hashing is not supported.
- MPLS strip is only supported for Layer 3 packets under the MPLS label stack.
- Use the new OFM-based MPLS stripping feature only if the deployment needs co-existence of MPLS stripping with any other type of header stripping such as VXLAN, iVXLAN, GRE, and ERSPAN headers.

  The existing MPLS stripping feature will continue to support MPLS stripping when co-existence is not needed with other stripping features.
- You must configure the TAP aggregation ACL with a redirect action on the ingress interface to forward the packet to the desired destination.
- Disable all Layer 3 and vPC features before you enable MPLS stripping.
- Static MPLS, MPLS segment routing, and MPLS stripping cannot be enabled at the same time.
- Only the ingress interfaces involved in MPLS stripping must have TAP aggregation enabled.
- Post MPLS strip, SMAC changes to switch mac (**show vdc**) and DMAC is set to **00:00:00:ab:cd:ef**.
- The egress interface where stripped packets will exit must be an interface that has VLAN 1 as an allowed VLAN. We recommend that you configure the egress interface as a trunk with all VLANs allowed by default.
- Stripping is based on IP PACL, and you cannot use MAC-ACL for stripping.
- During MPLS stripping, the incoming VLAN is not preserved.
-
- For every ACE (under an ACL associated with a particular VLAN) with a unique redirect port list, we allocate a hardware entry. The current hardware limit for the number of ACEs is 50 and you cannot configure more than 50 such ACEs.

Enable MPLS Stripping

Before you begin

Disable all Layer 3 and vPC features before you enable MPLS stripping.

Attach an ACL with the tap aggregation policy to the Layer 2 interface or port channel using the **mode tap-aggregation** command. For more information, see Attach TAP Aggregation Policy to Interface, on page 516 .

You can enable MPLS stripping globally.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Globally enable MPLS stripping using the command [**no**] **mpls strip**

Example:

```
switch(config)# mpls strip
```

The **no** form of this command disables MPLS stripping.

Step 3 Enable VLAN tagging on the packets coming from the redirect port using the command [**no**] **mpls strip mode dot1q**

Example:

```
switch(config)# mpls strip mode dot1q
```

The VLAN that needs to be tagged must be specified in the ingress port.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

Configure Incoming Port for VLAN Tag

The VLAN tag is derived from the incoming port configuration. The ingress/egress ports can either be ethernet or port channel.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Enter interface configuration mode for the specified interface using the command **interface** *type slot/port*

Example:

```
switch(config)# interface ethernet 1/26
 switch(config-if)#
```

Step 3 Change a Layer 3 interface to a Layer 2 interface using the command **switchport**

Example:

```
switch(config-if)# switchport
```

Note Make sure that the interface is a Layer 2 interface.

Step 4 Enter one of the following commands:

- [**no**] **ip port access-group** *access-list-name* **in**
- [**no**] **mac port access-group** *access-list-name* **in**

Example:

```
switch(config-if)# ip port access-group test in
      switch(config-if)# mac port access-group test in
```

Applies an IPv4 or MAC ACL configured with TAP aggregation to the interface. The **no** form of this command removes the ACL from the interface.

Step 5 Enter one of the following commands:

- [**no**] **ip port access-group** *access-list-name* **in**
- [**no**] **mac port access-group** *access-list-name* **in**

Example:

```
switch(config-if)# ip port access-group test in
 switch(config-if)# mac port access-group test in
```

Applies an IPv4 or MAC ACL configured with TAP aggregation to the interface. The **no** form of this command removes the ACL from the interface.

Step 6 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config-if)# copy running-config startup-config
```

This is an optional step.

Add and Delete MPLS Labels

The device can learn the labels dynamically whenever a frame is received with an unknown label on a TAP interface. You can also add or delete static MPLS labels.

Before you begin

Configure a TAP aggregation policy and attach the policy to an interface. For more information, see the *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide* .

You must configure the TAP aggregation ACL with a redirect action on the ingress interface to forward the packet to the desired destination.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Add the specified static MPLS label using the command **mpls strip label** *label*

Example:

```
switch(config)# mpls strip label 100
```

The 20-bit value of the label can range from 1 to 1048575.

Note This CLI is available for all the platform switches specified for the MPLS Stripping feature in the Guidelines and Limitations section, except for the following cloud scale platform switches:

- N9K-C93180YC-FX
- N9K-C93240YC-FX2
- N9K-C93180YC-FX3S
- N9K-C93600CD-GX

The [**no**] **mpls strip label** { *label* | **all** } command deletes the specified static MPLS label. The **all** option deletes all static MPLS labels.

Step 3 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure Destination MAC Address

You can configure the destination MAC address for stripped egress frames.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
   switch(config)#
```

Step 2 Specify the destination MAC address for egress frames that are stripped of their headers using the command **mpls strip dest-mac** *mac-address*

Example:

```
switch(config)# mpls strip dest-mac 1.1.1
```

The MAC address can be specified in one of the following four formats:

- E.E.E
- EE-EE-EE-EE-EE-EE
- EE:EE:EE:EE:EE:EE
- EEEE.EEEE.EEEE

Step 3 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

Configure MPLS Label Aging

You can define the amount of time after which dynamic MPLS labels will age out, if unused.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Specify the amount of time in seconds after which dynamic MPLS labels age out using the command **mpls strip label-age** *age*

Example:

```
switch(config)# mpls strip label-age 300
```

The range is from 61 to 31622400.

Step 3 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step.

MPLS Stripping Configuration Commands and Examples

Command Reference

The following table describes the command and its options for verifying MPLS stripping configuration.

Table 40: MPLS Strip Labels Command Options

Command	Purpose
show mpls strip labels [*label* **| all | dynamic | static]**	Displays information about MPLS labels. You can specify the following options: • *label* —Label to be displayed. • **all** —Specifies that all labels must be displayed. This is the default option. • **dynamic** —Specifies that only dynamic labels must be displayed. • **static** —Specifies that only static labels must be displayed.

To clear the MPLS stripping counters and label entries, perform these tasks:

Command	Purpose
clear mpls strip label dynamic	Clears dynamic label entries from the MPLS label table.
clear counters mpls strip	Clears all MPLS stripping counters.

Displaying MPLS Strip Labels

This example shows how to display all MPLS labels:

```
switch#
show mpls strip labels
MPLS Strip Labels:
Total      : 3005
Static     : 5
Legend:     * - Static Label
Interface - where label was first learned
Idle-Age  - Seconds since last use
SW-Counter- Packets received in Software
HW-Counter- Packets switched in Hardware

--------------------------------------------------------------------------------
```

```
                    Label    Interface      Idle-Age   SW-Counter       HW-Counter

-------------------------------------------------------------------------------
                    4096    Eth1/53/1            15            1              210
                    4097    Eth1/53/1            15            1              210
                    4098    Eth1/53/1            15            1              210
                    4099    Eth1/53/1             7            2              219
                    4100    Eth1/53/1             7            2              219
                    4101    Eth1/53/1             7            2              219
                    4102    Eth1/53/1            39            1              206
                    4103    Eth1/53/1            39            1              206
                    4104    Eth1/53/1            39            1              206
                    4105    Eth1/53/1             1            1              217
                    4106    Eth1/53/1             1            1              217
                    4107    Eth1/53/1             1            1              217
                    4108    Eth1/53/1            15            1              210
                    *   25000    None <User>          39            1              206
                    *   20000    None <User>          39            1              206
                    *   21000    None <User>           1            1              217
```

This example shows how to display only static MPLS labels:

```
                    switch(config)#
                    show mpls strip labels static
                    MPLS Strip Labels:
                    Total      : 3005
                    Static     : 5
                    Legend:     * - Static Label
                    Interface - where label was first learned
                    Idle-Age  - Seconds since last use
                    SW-Counter- Packets received in Software
                    HW-Counter- Packets switched in Hardware

-------------------------------------------------------------------------------
                    Label    Interface      Idle-Age   SW-Counter       HW-Counter

-------------------------------------------------------------------------------
                    *     300    None <User>         403           0               0
                    *     100    None <User>         416           0               0
                    *   25000    None <User>         869           0               0
                    *   20000    None <User>         869           0               0
                    *   21000    None <User>         869           0               0
```

How to clear MPLS stripping counters

The following example shows how to clear all MPLS stripping counters:

```
switch# clear counters mpls strip
switch# show mpls strip labels
MPLS Strip Labels:
    Total      : 15000
    Static     : 2
Legend:     * - Static Label
    Interface - where label was first learned
    Idle-Age  - Seconds since last use
    SW-Counter- Packets received in Software
    HW-Counter- Packets switched in Hardware
-------------------------------------------------------------------------------
     Label    Interface      Idle-Age   SW-Counter       HW-Counter
-------------------------------------------------------------------------------
```

```
 4096    Eth1/44              15            0               0
 8192    Eth1/44              17            0               0
12288    Eth1/44              15            0               0
16384    Eth1/44              39            0               0
20480    Eth1/44              47            0               0
24576    Eth1/44               7            0               0
28672    Eth1/44               5            0               0
36864    Eth1/44               7            0               0
40960    Eth1/44              19            0               0
45056    Eth1/44               9            0               0
49152    Eth1/44              45            0               0
53248    Eth1/44               9            0               0
```

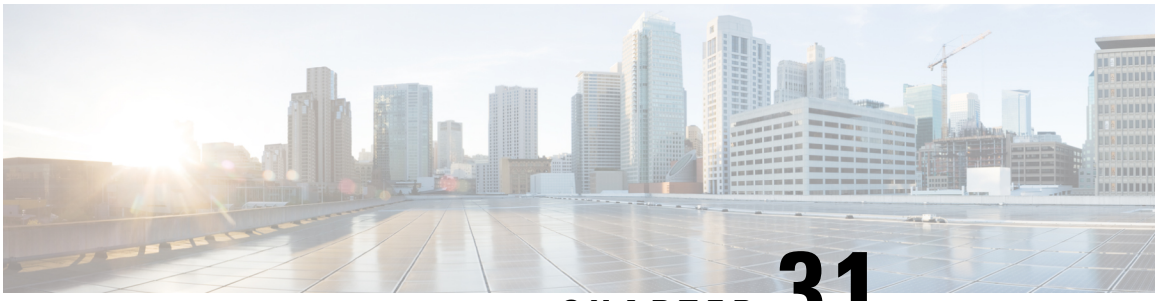
How to Add Static MPLS labels

```
switch# configure terminal
switch(config)# mpls strip label 100
switch(config)# mpls strip label 200
switch(config)# mpls strip label 300
```

Related Documents

This topic provides references to related documents and topics for further information.

Related Topic	Document Title
IP ACLs	*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*
MAC ACLs	*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*
Port-channel symmetric hashing	*Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*
Remote monitoring (RMON)	Configuration, on page 287
Switched port analyzer (SPAN)	Switched Port Analyzer, on page 367
Troubleshooting	*Cisco Nexus 9000 Series NX-OS Troubleshooting Guide*



CHAPTER 31

Packet truncation

Packet truncation is a software feature that allows a Cisco Nexus 9000 Series switch to shorten packets to a user-defined size as they exit an interface. You can control the Cyclic Redundancy Check (CRC) value of truncated packets, streamline monitoring by truncating packets at the egress port, and save resources.

Packet truncation lets you perform truncation directly on the egress port that connects to your monitoring tool. This approach simplifies your workflow and saves resources.

For more information, see Packet Truncation.

In typical Network Data Broker (NDB) environments, monitoring tools receive traffic using SPAN (Switched Port Analyzer) sessions. When full packet payloads are not required, SPAN truncation is often used. However, this method requires redirecting traffic through additional ports configured with a Port Access Control List (PACL), consuming the port capacity.

The packet truncation feature enables you to truncate packets at the egress port. This optimizes network monitoring traffic, reduces infrastructure costs, and addresses the operational overhead associated with traditional packet capture methods.

Components of packet truncation

Packet truncation includes three main components:

- **Egress Port Level Truncation**

  This function is available on CloudScale switches. Administrators can define a maximum size for packets on a specific egress port. Packets larger than this size are truncated. The configuration is managed using the CLI and is integrated into the Cisco NX-OS Device Management Engine (DME).

- **CRC Stomping Control**

  On the Cisco Nexus switches, packet truncation invalidates the original CRC value. The switch replaces the packet with a bad CRC causing the receiving peer or monitoring tool to discard the packet. This feature provides a CLI command to disable CRC stomping.

- **CRC Check Control (Legacy Hardware)**

  On the Cisco Nexus FX, FX2, and FX3 Series switches, a system-level command is available to disable the CRC check for truncated packets. This allows packets with CRC anomalies after truncation, to be forwarded successfully to monitoring tools.

Configure and manage the egress packet truncation

Use these commands to configure and manage packet truncation at the egress port.

Command	Description
packet-truncation size *size_in_bytes*	Enable packet truncation on an egress interface and set the maximum packet size in bytes.
[**no**] **packet-truncation crc-stomp**	Disable CRC stomping on truncated packets to ensure the peer device receives a valid CRC.The option to enable or disable packet-truncation crc-stomp is not supported on Nexus 9300-FX, FX2, and FX3 platforms.
[**no**] **system packet-truncation crc-check**	Disable the CRC check for truncated packets across the system.

Guidelines and limitations for packet truncation

Follow these guidelines when you use packet truncation.

- The packet truncation feature can be enabled only after enabling feature tap-aggregation feature.
- The **packet-truncation size** command is configurable only under physical interface and port-channel interfaces. It is *not* configurable on the Layer 3 interfaces and SVI interfaces.
- The [**no**] **packet-truncation crc-stomp** is configurable only under physical interface and port-channel interfaces. It is *not* configurable on the Layer 3 interfaces and SVI interfaces.

  It is *not* configurable on Cisco Nexus FX, FX2 and FX3 Series switches.
- The [**no**] **system packet-truncation crc-check** command is configurable at the global level. It allows crc-stomped packets from the previous hop and forwards the packets to next hop.
- Packet truncation is supported on the Cisco Nexus CloudScale switches.

  Packet truncation is *not* supported on these Cisco Nexus switches.
  - Cisco Nexus 9500 Series switches
  - Cisco Nexus SiliconOne switches
- If packet-truncation is configured under interfaces (such as Po and Eth), it can affect the control packets such as LLDP, LACP, STP, and CDP egressing out of those interfaces depending on the size configured. Lower packet-truncation size can corrupt the egress protocol packets, resulting in neighborship formation or ports going to suspended state. Beginning with NX-OS Release 10.6(3)F, a new command executed at the interface or global level allows you to exclude the control packets from being truncated.

- Configure packet truncation at the egress ports task, on page 531
- Configure packet truncation with CRC stomping disabled, on page 531
- Configure packet truncation to exclude control packets, on page 532
- Configure system-wide packet truncation CRC check, on page 534
- Verify packet truncation , on page 535

Configure packet truncation at the egress ports task

Use this procedure to configure an interface to truncate all egress packets to a specific size.

Procedure

Step 1 Enter global configuration mode.

Example:

```
switch# configure terminal
```

Step 2 Specify the interface to configure and enter interface configuration mode.

Example:

```
switch(config)# interface ethernet 1/1
```

Step 3 Enable packet truncation of egress packets to a specific size on the interface.

Example:

```
switch(config)# packet-truncation size 128
```

Step 4 Exit interface configuration mode.

Example:

```
switch(config)# exit
```

Example

This is a sample configuration example to truncate packets to a specific size of 128 bytes.

```
switch# configure terminal
switch(config)# interface ethernet 1/1
switch(config-if)# packet-truncation size 128
switch(config-if)# end
```

Configure packet truncation with CRC stomping disabled

Use this procedure to disable CRC stomping on Cisco Nexus GX, GX2, and HX Series switches.



Note If the Cisco Nexus GX, GX2, and HX Series switches receives a truncated packet with a bad CRC from the previous hop, the packet must be further truncated on this switch to a smaller size to utilize the functionality of this command and avoid CRC stomping.

Procedure

Step 1 Enter global configuration mode.

Example:

```
switch# configure terminal
```

Step 2 Specify the interface to configure and enter interface configuration mode.

Example:

```
switch(config)# interface ethernet 1/1
```

Step 3 Enable packet truncation of egress packets to a specific size on the interface.

Example:

```
switch(config)# packet-truncation size 128
```

Step 4 Disable CRC stomping to ensure truncated packets are sent with a valid CRC.

Example:

```
switch(config)# no packet-truncation crc-stomp
```

Step 5 Exit interface configuration mode.

Example:

```
switch(config)# exit
```

Example

The example shows how to configure Ethernet interface 1/2 to truncate all egress packets to 256 bytes and ensures they are sent with a valid CRC.

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# packet-truncation size 256
switch(config-if)# no packet-truncation crc-stomp
switch(config-if)# end
```

Configure packet truncation to exclude control packets

In earlier releases, including 10.6(1)F and 10.6(2)F, packet truncation on interfaces such as PO and Ethernet can corrupt control packets (LLDP, LACP, STP, CDP) if set too low, caused neighborship or port issues. However, from NX-OS Release 10.6(3)F you can exclude control packets from truncation either at the system level or at the interface level.

Configure packet truncation to exclude control packets at system level

Use this procedure to exclude control truncated packets at system level.

Procedure

Step 1 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
switch# configure terminal
```

Step 2 Use the **system packet-truncation exclude control-packets** command to enable the exclusion of control packets from packet truncation. This prevents essential control packets from being truncated while still applying truncation to data packets.

Example:

```
switch(config)# system packet-truncation exclude control-packets
```

Step 3 Use the **exit** command to exit the global configuration mode.

Example:

```
switch(config)# exit
```

This is a sample configuration example to exclude control packets from being truncated.

```
switch# configure terminal
switch(config)# system packet-truncation exclude control-packets
switch(config)# exit
```

Configure packet truncation to exclude control packets on an interface

Use this procedure to configure an interface to exclude control truncated packets.

Procedure

Step 1 Use the **configure terminal** command to enter the global configuration mode.

Example:

```
switch# configure terminal
```

Step 2 Use the **interface ethernet slot/port** command to specify the interface to configure and enter interface configuration mode.

Example:

```
switch(config)# interface ethernet 1/1
```

Step 3 Enable packet truncation of egress packets to a specific size on the interface. It is mandatory to enable packet truncation size before enabling exclude control packets under an interface.

Example:

```
switch(config-if)# packet-truncation size 128
```

Step 4 Enable exclusion of control packets from packet truncation. This prevents essential control packets from being truncated while still applying truncation to data packets.

Example:

```
switch(config-if)# packet-truncation exclude control-packets
```

Step 5 Exit interface configuration mode.

Example:

```
switch(config-if)# exit
```

This is a sample configuration example to exclude control packets from being truncated.

```
switch(config)# interface ethernet 1/1
switch(config-if)# packet-truncation size 128
switch(config-if)# packet-truncation exclude control-packets
switch(config-if)# exit
```

Configure system-wide packet truncation CRC check

Use this task to enable system-wide CRC check and to configure packet truncation for egress ports.

Procedure

Step 1 Enter global configuration mode.

Example:

```
switch# configure terminal
```

Step 2 Disable the system-wide CRC check for truncated packets. This global command accepts all ingress packets with CRC errors instead of dropping it.

Example:

```
switch(config)# no system packet-truncation crc-check
```

Step 3 Specify the interface to configure and enter interface configuration mode.

Example:

```
switch(config)# interface ethernet 1/1
```

Step 4 Configure the interface to truncate all egress packets to 128 bytes.

Example:

```
switch(config)# packet-truncation size 128
```

Step 5 Exit interface configuration mode.

Example:

```
switch(config)# exit
```

Example

The example shows how to disable the system-wide CRC check and configure the Ethernet interface 1/3 to truncate all egress packets to 128 bytes:

```
switch# configure terminal
switch(config)# no system packet-truncation crc-check
switch(config)# interface ethernet 1/3
switch(config-if)# packet-truncation size 128
switch(config-if)# end
```

Verify packet truncation

To verify packet truncation settings on Cisco Nexus 9000 switches, use these commands and review the outputs.

Interface verification

Use the **show running-config interface** command to verify packet truncation settings on a specific interface:

```
Switch# show running-config interface ethernet 1/2
...
interface Ethernet1/2
packet-truncation size 256
no packet-truncation crc-stomp
```

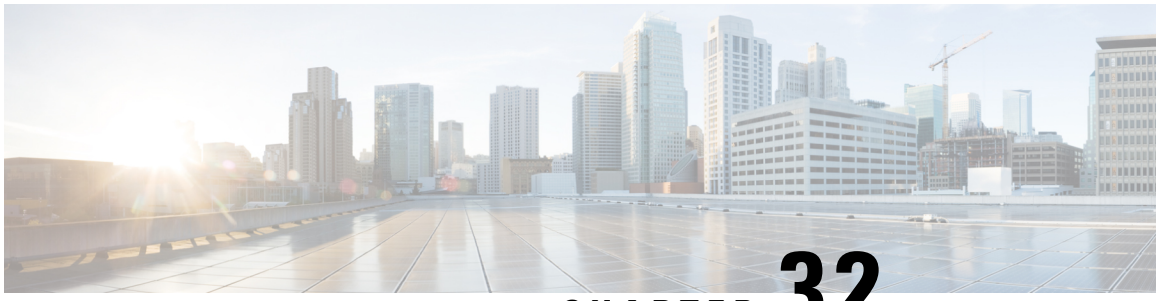
This command displays the packet truncation size and indicates if CRC stomp is disabled for the selected interface.

System verification

Use the **show running-config** command to check global packet truncation CRC settings on the Cisco Nexus 9000 FX, FX2 and FX3 Series switches:

```
Switch# show running-config | include crc-check
no system packet-truncation crc-check
```

This command disables the system level check for bad CRC packets and forwards it to networking monitoring tool for analysis.



CHAPTER 32

Configuring Deduplication

This chapter describes how to configure deduplication on Cisco NX-OS devices.

This chapter contains the following sections:

- Deduplication concept and models, on page 537
- Guidelines and limitations for deduplication, on page 538
- Configure deduplication, on page 539
- Monitor commands for deduplication, on page 540

Deduplication concept and models

Deduplication is a feature that

- removes duplicate data to ensure only unique data packets are processed and stored
- preventss unnecessary storage consumption by eliminating redundant copies of traffic, and
- supports two operational models.

Due to the continuous rise in data flow, navigating in the networks and the data processed in different applications leads to data duplication both in source and target. For efficient data management, security, and storage you can use deduplication to remove duplicate data.

The deduplication feature eliminates the duplicate traffic which is received from traffic analysers or data storage tools. It identifies duplicate flows which are transferred in the Nexus Data Broker (NDB) switch.

Deduplication feature can be used to identify and eliminate duplicate data flows as they traverse data broker switches. By removing redundant copies of traffic—often generated by traffic analyzers or data storage tools—deduplication ensures that only unique data packets are processed and stored.

Deduplication supports two model types:

- **Inline deduplication Model (Interface Mode)**: You can configure the inline model on existing NDB switches. This eliminates duplicate packets using interfaces. Initial interface packets are forwarded and the duplicate packets from other interfaces are dropped.
- **Deduplication on a Stick Model (VLAN Mode)**: In this model, the packet flow is bound to a specific VLAN. The packet flow is permitted on the first VLAN, and duplicate flows on other VLANs are restricted. The packets are tagged with specific VLAN when the flow enters the data broker switches.

Each packet is tagged with a unique VLAN, the packets with VLAN tag are forwarded to the deduplication switch, and duplicate flow is removed on that switch.

Guidelines and limitations for deduplication

This topic lists the classified categories of the guidelines and limitations for configuring deduplication.

Supported features

- You can configure deduplication for inner packet flows only.
- TCP and UDP flows are supported for flow deduplication.
- Deduplication is supported for local SPAN and Optical TAP flows sessions.
- Deduplication can be performed for VXLAN and GRE tunnel packets with a single encap.
- Dynamic aging is supported only for 128k flows.

Unsupported features

- Deduplication stick model (VLAN model) is not supported for tunnel termination, and Q-in-Q ports.
- Deduplication is not supported for IPv6 and multicast flows.

Platform support

- Deduplication is supported for Cisco Nexus 9300-FX2/FX3 and 9300-GX/GX2 platorm switches.
- Deduplication is not supported for:
  - Cisco Nexus 9500 platorm switches
  -
  - Cisco N9K-C9408 switch

ISSU

During ISSU, deduplication is disabled by default. Post ISSU all flows are cleared and refreshed.

Flows

- You cannot delete duplication flow for short lived flows.
- Flow packets from different VRFs with similar 5 tuples cannot be deleted. As deduplication is for the flows performed using 5-tuple of a packet. It cannot identify VRF packets.
- For deduplication to function appropriately on tunnel traffic, ensure that you enable flow terminate.

Duplicate packets

You cannot configure deduplication for duplicate packets on the same interface such as ERSPAN. It may terminate more than a single ERSPAN session on an interface or SVI. This impacts the copy of same flow ends up on the same interface or VLAN. A switch cannot differentiate different ERSPAN sessions for deduplication.

Configure deduplication

Deduplication feature can be used to remove duplicate copies of the data flow when they are going through switches in Nexus Data Broker (NDB).

Procedure

Step 1 Enter global configuration mode using the **configure terminal** command.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Enable flow-deduplication feature on a switch using the **tap-aggregation flow-deduplication** command.

Example:

```
switch(config)# tap-aggregation flow-deduplication
switch(config-dedup)#
```

Step 3 (Optional) Configure absolute timer for deduplication of packet flows using the **absolute-timer***time in minutes* command.

The interval range is from 0 to 1440 minutes.

Example:

```
switch(config-dedup)# absolute-timer 10
switch(config-dedup)#
```

Step 4 (Optional) Configure dynamic timer for deduplication of packet flows using the **dynamic-timer***time in milli seconds* command.

The interval range is from 0 to 300000 milliseconds.

Example:

```
switch(config-dedup)# dynamic-timer 2000
switch(config-dedup)#
```

Step 5 Configure deduplication on required mode using the **mode** {**vlan**|**interface**} command.

Note Ensure to save configuration and reload the switch to configure deleting duplication on the switch.

Example:

```
switch(config-dedup)# mode interface
switch(config-dedup)#
```

Step 6 Clear deduplication on the required slot or module using the **clear hardware deduplication statistics**{**slot**|**module in number**} command.

The slot or module number range is from 1 to 30.

What to do next

Ensure that you reload switch after configuring deduplication, for the configuration changes to be effective.

Monitor commands for deduplication

Use the show commands listed in this section to monitor dedeplication. The examples display sample outputs.

- **show hardware deduplication summary**

```
switch# show hardware deduplication summary
slot 1
=======
Deduplication      : Enabled
Dedup Mode         : Interface
Dynamic timer                : 200000 milliseconds
Absolute timer               : 5 minutes
Max Supported Flows          : 240K
Total number of learned flows : 240000
Total number of dropped bytes : 65698869600
```

- **show hardware deduplication detail**

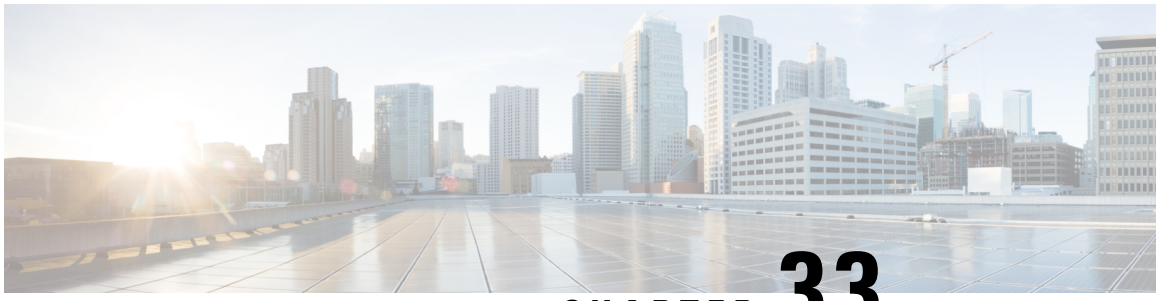
```
switch# show hardware deduplication detail
slot1
================================================================================
   Dedup Flows
================================================================================
SourceIP  Destination IP  Ports(Src:Dst) Protocol    Interface     Learn-time
================================================================================
192.0.2.10  198.51.100.10  3000 :3001  6  Eth1/1  07/28/2023 11:47:09.532376
192.0.2.20  198.51.100.11  15000:15001 17      Eth1/1  07/28/2023 11:47:09.532229
192.0.2.30  198.51.100.12  1841 :1842  6  Eth1/1  07/28/2023 11:47:09.532340
192.0.2.40  198.51.100.20  2000 :2001  6  Eth1/1  07/28/2023 11:47:09.532428
192.0.2.50  198.51.100.21  4000 :4001  6  Eth1/23 07/28/2023 11:47:09.532133
```

- **show hardware deduplication age-history**

```
switch#show hardware deduplication age-history
slot 1
================================================================================
    Dedup Flows
================================================================================
Source    Destination     Ports  Protocol   Interface Timer    Learn-Time    Aged-Time

IP   IP       (Src:Dst)
================================================================================
192.0.2.50 203.0.113.20   4000:4001  6  Eth1/17 Dynamic   08/05/2023 2:24:49.26020
  08/05/2023 12:33:29.21904
192.0.2.10 203.0.113.10   3000:3001      6  Eth1/27 Dynamic   08/05/2023 12:24:49.126246
    08/05/2023 12:33:29.21945
192.0.2.20 203.0.113.11  15000:15001     17 Eth1/5 Dynamic    08/05/2023 12:24:49.26070
     08/05/2023 12:33:29.21957
192.0.2.40 203.0.113.21   2000 :2001      6  Eth1/5 Dynamic    08/05/2023 12:24:49.26115
```

```
     08/05/2023 12:33:29.21969
192.0.2.30 203.0.113.12   1841 :1842      6  Eth1/17 Dynamic   08/05/2023 12:24:49.25949
     08/05/2023 12:33:29.21979
```



CHAPTER 33

MPLS Access Lists

- Configure MPLS access lists, on page 543
- Verification of MPLS access lists configuration, on page 544
- Configuration example for MPLS access lists, on page 544

Configure MPLS access lists

MPLS access lists enable filtering of MPLS packets based on MPLS label and sending filtered packets to configured redirect interfaces.

Procedure

Step 1 Enter global configuration mode using the **configure terminal** command.

Example:

```
switch# configure terminal
switch(config)#
```

Step 2 Enable parsing of MPLS packets using the [**no**]**install feature-set mpls** command.

This is mandatory to filter MPLS packets based on MPLS label.

Example:

```
switch(config)# install feature-set mpls
switch(config)# feature-set mpls
switch(config)# feature mpls segment-routing
```

Step 3 Configure mpls-access list with filtering based on incoming outer MPLS label using the **mpls access list mpls-acl** command.

In the example, MPLS packets with incoming label 1600 are matched and redirected to Ethernet1/15.

Example:

```
switch(config)# mpls access list mpls-acl
switch(config-mpls-acl)# 10 permit mpls 1600 any redirect Ethernet1/15
```

Step 4 (Optional) Copy the running configuration to the startup configuration using the **copy running-config startup-config** command.

Example:

```
switch(config)# copy running-config startup-config
```

Verification of MPLS access lists configuration

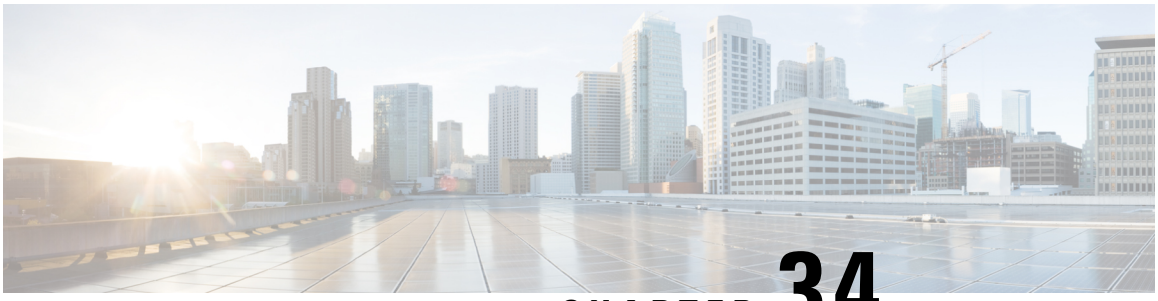
Use the show command listed in the table to verify the output that displays information about the MPLS access list configuration.

Command	Purpose
show mpls access lists	Displays information about MPLS access lists.

Configuration example for MPLS access lists

This example shows how to configure MPLS access lists.

```
switch# configure terminal
switch(config)# install feature-set mpls
switch(config)# feature-set mpls
switch(config)# feature mpls segment-routing
switch(config)# mpls access list mpls-acl
switch(config-mpls-acl)# 10 permit mpls 1600 any redirect Ethernet1/15
switch(config)# copy running-config startup-config
```



CHAPTER 34

Header Stripping for Nexus Data Broker

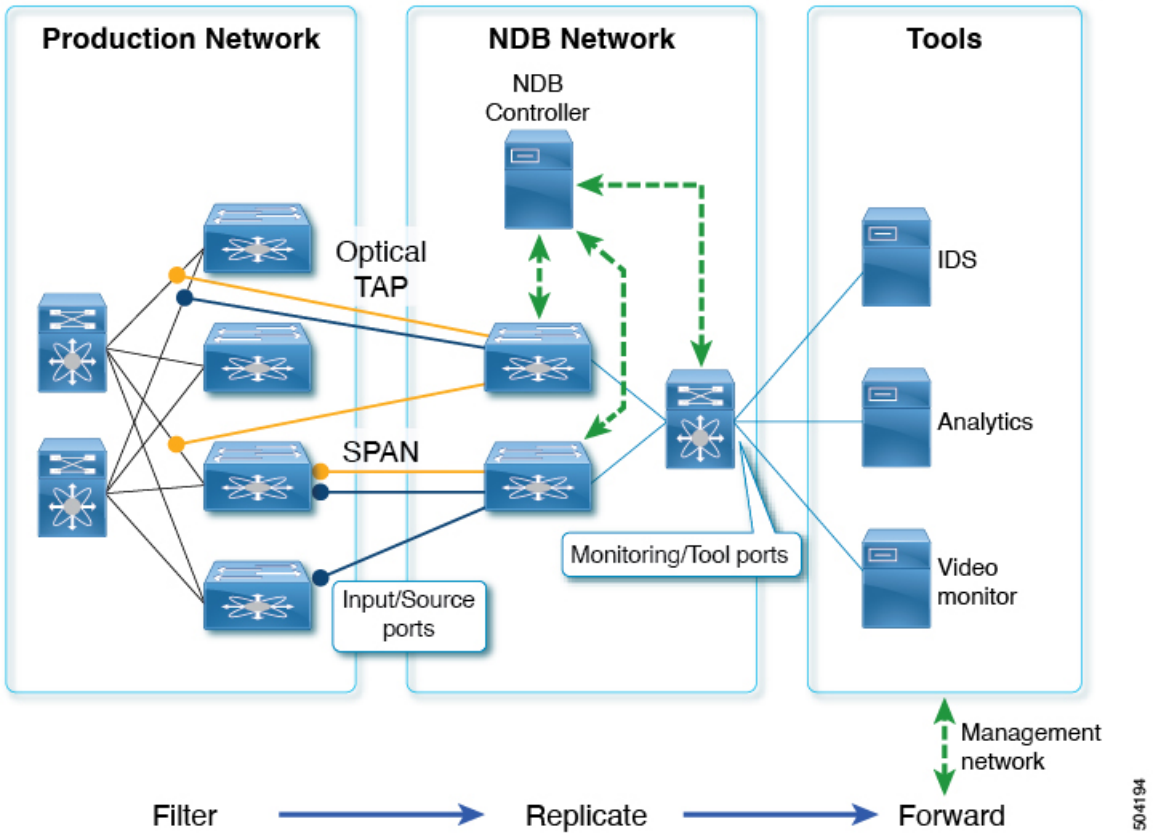
- Overview, on page 545
- Guidelines and Limitations, on page 547
- VXLAN and iVXLAN Header Stripping, on page 548
- ERSPAN Header Stripping, on page 552
- GRE Header Stripping, on page 556
- MPLS Header Stripping, on page 559

Overview

Cisco Nexus Data Broker (NDB) is a solution that builds scalable packet broker network solutions, providing a software-defined approach for monitoring both out-of-band and inline network traffic using Cisco Nexus Dashboard Data Broker controller software and Cisco Nexus switches.

- NDB enables packet monitoring for performance monitoring, intrusion detection, and compliance checking.
- Header stripping is performed on NDB switches, allowing traffic to be filtered, replicated, and stripped of headers before being forwarded to monitoring tools.
- Input/source ports are where header stripping occurs, and monitoring/tool ports connect directly to the tools.

Figure 10: NDB Centralized Deployment Model



Header Stripping and Feature Benefits

The reasons for removing the header are as follows:

- Some monitoring tools do not understand an encapsulated packet.
- Presence of an additional header skews the analytics data.
- Addition of a header adds to the packet size, hampering the optimization of the amount of data that is sent to and processed by the tools.

The benefits of the packet header or label stripping feature of Cisco Nexus Data Broker switch are as follows:

- Enable Multiprotocol Label Switching (MPLS) label stripping
- Native support for VXLAN header stripping from copy traffic
- Support for Generic Route Encapsulation (GRE) header stripping
- Q-in-Q VLAN header stripping at egress

NDB aligns the legacy VXLAN, IVXLAN, ERSPAN, GRE, and MPLS stripping functionality to the Overlay Forwarding Manager (OFM) based model. The OFM hosts the command line interface (CLI) for header stripping functionality.

This chapter contains the following sections:

- VXLAN and iVXLAN Header Stripping, on page 548
- ERSPAN Header Stripping
- GRE Header Stripping
- MPLS Header Stripping

Guidelines and Limitations

This topic lists the guidelines and limitations applicable to all header stripping features.

General Guidelines and Limitations

- A maximum of 500 flow terminate interfaces are supported across all tunnel-profiles with various encapsulation types such as VxLAN, iVxLAN, GRE, and MPLS. For ERSPAN, the maximum flow terminate interfaces supported is 31.
- The co-existence can be on the same interface or different interfaces.
- The legacy MPLS stripping feature and OFM stripping features are mutually exclusive.
- After performing non-disruptive ISSU from an earlier release to Cisco NX-OS Release 10.2(3)F and performing any header stripping functions, if dot1q tunnel VLAN_tag is missing or set to vlan_id=1, then remove and add the port ACL from L2 interfaces for that particular stripping-enabled interface.
- If no VLAN is configured on an interface, but the switchport mode dot1q-tunnel command is configured on that interface, then stripped packets will have VLAN=1 by default.
- In a scenario where incompatible OFM commands are present in the show running command output, and disruptive ISSU from Cisco NX-OS Release 10.2(3)F to an earlier release is done, wherein OFM commands were not supported in the earlier NX-OS version, then appropriate errors are displayed. However, the show incompatibility command does not flag such errors for OFM-related incompatibility commands.
- The OFM-based GRE, ERSPAN, and MPLS stripping features are supported only on TORs, not on line cards.
- As part of the encapsulation (iVXLAN, VXLAN, GRE, MPLS, ERSPAN), the following restrictions are common:
  - Two or more tunnel-profiles cannot have the same encapsulation-type.
  - OFM-based header stripping features are not supported when feature tunnel is enabled.

Release Specific Guidelines

Beginning with Cisco NX-OS Release 10.2(3)F

- Traffic with IPv6 inner packet is supported for all stripping functions.
- The MPLS stripping using the OFM model co-exists with the other stripping features. However, the existing MPLS stripping feature will continue to support MPLS stripping when co-existence is not needed with other types of stripping features.

- ERSPAN coexistence on the same interface is supported. However, this is supported on 9300-FX2 and later platforms only.

VXLAN and iVXLAN Header Stripping

Overview

Nexus Data Broker (NDB) VXLAN and iVXLAN header stripping is a feature that enables switches to remove VXLAN and iVXLAN headers from packets upon receipt.

- It allows switches to terminate VXLAN and iVXLAN packets.
- It strips VXLAN and iVXLAN headers from incoming packets.
- It facilitates packet processing in scenarios involving TAP and SPAN configurations.

Scenarios for Header Stripping

NDB switches receive packets in the following scenarios:

- Test Access Point (TAP) ports between spines and leaf are placed on the Fabric Links in the ACI fabric.
- Switched Port Analyzer (SPAN) sessions are configured, or TAPs placed in the VXLAN overlay network.

Supported PIDs

Guidelines and Limitations

Post VXLAN and iVXLAN Header Strip Guidelines

- The interface will allow slapping Q-in-Q VLAN on inside packet.
- Packet CRC will be properly performed.
- Inside packets will be allowed to filter using ingress port ACLs.

Not Supported Features

VXLAN and iVXLAN strip is not supported if the following features are enabled:

- NV overlay
- VN-segment-vlan
- Legacy MPLS strip and tap-aggregation

Release Specific Guidelines

Beginning with Cisco NX-OS Release 10.2(1)F

- The VXLAN and iVXLAN stripping features are supported on Cisco Nexus 9364C and 9300-EX, 9300-FX, 9300-FX2, 9500-EX, and 9500-FX line cards.

Beginning with Cisco NX-OS Release 10.2(2)F

- The VXLAN and iVXLAN stripping features are supported on Cisco Nexus 9300-GX and 9300-GX2 platform switches.

Beginning with Cisco NX-OS Release 10.2(3)F

- VXLAN strip is supported on Cisco N9K-C93180YC-FX3 and N9K-C93108TC-FX3P platform switches.
- A maximum of 5 tunnel-profiles are supported.

Beginning with Cisco NX-OS Release 10.2(4)M

- The iVXLAN stripping feature is supported on Cisco N9K-C93180YC-FX3 and N9K-C93108TC-FX3P platform switches.

Platform Specific Guidelines

- You must be able to strip VXLAN, and iVXLAN headers without being PTEP/VTEP.
- VXLAN header strip is enabled per port.
- Ports must be able to manage both tunneled and non-tunneled packets.
- Layer 2 switch port mode trunk or Layer 2 PO interfaces must be able to strip the VXLAN header.
- Ensure that the Tap-ACL contains proper ACE with redirect keyword, where the redirect interfaces are pointing toward the egress/analyzer ports, else the packet will be flooded back on the same ingress port.
- OFM enables VXLAN strip capability for standard ISSU and LXC-ISSU.
- A maximum of 4 tunnel-profiles can be created on a switch, one per type of encapsulation.
- A maximum of 12 redirect interfaces (prior to Release 10.2(1)) and 32 redirect interfaces (Release 10.2(1) and later) can only be configured in a single ACE of the TAP aggregation policy.
- For Cisco Nexus 9300-GX platform switches, post VXLAN strip, L2 header addresses are re-written as follows: Source MAC as VDC MAC address and Destination MAC as 000000abcdef.

Supported Features

- VXLAN header strip is supported when VXLAN underlay is V4.
- VXLAN stripping is supported when the default UDP value is used.
- The following switches support VXLAN and iVXLAN header stripping feature from the mentioned releases:
  - N9K-C9348GC-FX3 – 10.4(1)F
  - N9K-C9332D-H2R – 10.4(1)F
  - N9K-C93108TC-FX3 – 10.4(2)F
  - N9K-C93400LD-H1 – 10.4(2)F

- N9K-C92348GC-FX3 – 10.5(2)F
- N9K-X9736C-FX3 line card - 10.5(2)F

Configure Nexus Data Broker Termination

The following steps outline the termination of NDB for VXLAN. The same procedure is followed for iVXLAN header strip.

Note To change encapsulate tunnel type from VXLAN to iVXLAN or vice versa, the configured tunnel must be removed using no encapsulate CLI.

Note Ensure that the below CLIs are configured to enable stripping of VXLAN or iVXLAN on interfaces:

- destination any
- encapsulation vxlan
- flow terminate interface add Ethernet 1/1

If any of the above CLIs are missing, stripping of VXLAN or iVXLAN will not happen on the ports specified in flow term CLI.

Procedure

Step 1 Enter global configuration mode using the command **configure terminal** and enable feature ofm using the command **feature ofm**

Example:

```
switch# configure terminal
```

Example:

```
switch (config)# feature ofm
```

Step 2 Enable static VXLAN tunnels using the command **tunnel-profile profile-name** and set appropriate encapsulation type for the tunnel profile using the command **encapsulation vxlan**

Example:

```
switch(config)# tunnel-profile vtep_vxlan_term
                        switch(config-tnl-profile)#
```

Example:

```
switch(config-tnl-profile)# encapsulation vxlan
                        switch(config-tnl-profile)#
```

Step 3 Set required destination for the tunnel profile using the command **destination any**

Example:

```
switch(config-tnl-profile)# destination any
```

Step 4 Add ethernet1/1 to the flow term list (if the **no flow terminate interface** command was configured) using the command **flow terminate interface ethernet 1/1** and remove Ethernet 1/1 port only using the command **flow terminate interface remove ethernet 1/1**

Example:

```
switch(config-tnl-profile)# flow terminate interface ethernet 1/1
```

Example:

```
switch(config-tnl-profile)# flow terminate interface remove ethernet 1/1
```

Step 5 Add e1/2, e1/3, e1/4, e1/5 to an existing list of flow terminate interfaces using the command **flow terminate interface add ethernet 1/2-5**

Example:

```
switch(config-tnl-profile)# flow terminate interface add ethernet 1/2-5
```

Note While adding flow terminate interface, CLI doesn't check whether L2 port interface exists or enabled. For example, e1/10 is a non-breakout mode. CLI allows interface e1/10/1-4 to add for flow terminate list. When e1/10 is a breakout, VXLAN header strip feature functions.

Step 6 Add port channel 100-110 to old list uisng the command **flow terminate interface add port-channel 100-110**

Example:

```
switch(config-tnl-profile)# flow terminate interface add po100-110
```

New list will be e1/10-11 and po100-110.

Step 7 Remove all flow and terminate interfaces from profile uisng the command **no flow terminate interface** and enable feature tap-aggregation using the command **feature tap-aggregation**

Example:

```
switch(config-tnl-profile)# no flow terminate interface
```

Example:

```
switch(config)# feature tap-aggregation
```

Step 8 Create an IPACL and enters the IP access list configuration mode using the command **ip access-list** <access-list name> and create an IP ACL rule that permits traffic to be redirected per its conditions using the command **[no] permit** protocol source destination redirect interfaces

Example:

```
switch(config)# ip access-list test
                        switch(config-acl)#
```

Example:

```
permit ip any any redirect interface ethernet 1/1, ethernet 1/19
```

The no version of this command removes the permit rule form the policy.

Note When you enter an interface for the TAP aggregation policy, do not abbreviate it. When you enter a list of interfaces, separate them with commas, but no spaces.

Step 9 Apply the port access list to the ERSPAN strip/terminating port using the command **ip port access-group** <access-group name> **in**

Example:

```
configure terminal
                        interface Ethernet 1/32
                        ip port access-group test in
```

The following example shows VXLAN and iVXLAN header stripping, the procedure is same for iVXLAN:

```
switch(config-tnl-profile)# show run ofm
show running-config ofm
feature ofm
tunnel-profile vxlan1
encapsulation vxlan
destination any
flow terminate interface add port-channel101
flow terminate interface add Ethernet1/1

tunnel-profile vxlan2
encapsulation ivxlan
destination any
flow terminate interface add port-channel101
flow terminate interface add Ethernet1/1
switch(config-tnl-profile)#
switch(config-tnl-profile)# show tunnel-profile
Profile : vxlan1
Encapsulation : Vxlan
State : UP
Destination : Any
Terminate Interfaces : 2
Terminate List : port-channel101 Ethernet1/1
Profile : vxlan2
Encapsulation : iVxlan
State : UP
Destination : Any
Terminate Interfaces : 2
Terminate List : port-channel101 Ethernet1/1
switch(config-tnl-profile)#
```

ERSPAN Header Stripping

Overview

ERSPAN header stripping is a feature that removes the ERSPAN header from incoming ERSPAN packets on NX-OS or NDB switches and forwards the packets inline based on ACL configuration.

- It strips ERSPAN headers from incoming packets.
- It operates inline on NX-OS and Nexus Data Broker switches.
- It redirects packets to outside servers based on ACL configuration.

Supported PIDs

Beginning with Cisco NX-OS Release 10.2(1)F, ERSPAN header stripping is supported on Cisco Nexus 9300-FX2, 9300-FX3, 9300-GX, and 9300-GX2 platform switches. However, this feature is only supported on TOR switches.

Guidelines and Limitations

General Limitations

- The ERSPAN header stripping feature is supported only on Cisco Nexus 9300-FX2, 9300-FX3, 9300-GX, and 9300-GX2 platform switches, and only on TOR switches.
- The incoming port must be a layer 2 port, with connectivity to layer 3 through SVI.
- Strips all the incoming ERSPAN headers on the terminating port.
- The incoming port must be a layer 2 port, but its connectivity to layer 3 must be through SVI.
- ERSPAN destination session and ERSPAN stripping cannot co-exist.
- The total number of terminating ports including port channel members cannot be more than 31.
- Mode tap-agg should not be configured for this feature.
- Tunnel profile for all ERSPAN ID is supported. Termination of specific ERSPAN session ID is not supported. Traffic with any ERSPAN session ID will be terminated at the termination node.

General Guidelines

- Only 1 tunnel profile per node is supported.
- A maximum of 31 flow terminate interfaces are supported on tunnel-profile with encap type: ERSPAN.
- The ERSPAN header stripping feature is supported on Cisco Nexus 9300-FX2, 9300-FX3, 9300-GX, and 9300-GX2 platform switches. Note that this feature is only supported on TOR switches.
- You need to enable ERSPAN stripping on the port so that ERSPAN strip/redirect works properly. Do not send ERSPAN traffic on ports where other strips are enabled.
- Strips all the incoming ERSPAN headers on the terminating port.
- This feature works only when OFM tunnel profiles and ACL redirect are configured and when port ACL is applied to the layer 2 terminating port.
- There can be only one tunnel profile for ERSPAN encapsulation on the switch.
- Appropriate tcam needs to be carved to use port acl, for example, **tcam region ing-ifacl** should be used for carving.

Configure ERSPAN Header Stripping

The following steps outline the configuration for ERSPAN header stripping.



Note Ensure that the below CLIs are configured to enable stripping of ERSPAN on interfaces:

- encapsulation erspan
- erspan session-id all
- flow terminate interface add e1/16

If any of the above CLIs are missing, stripping of ERSPAN does not happen on the ports specified in flow term CLI.

Procedure

Step 1 Enter global configuration mode uisng the command **configure terminal**

Example:

```
switch# configure terminal
```

Step 2 Enable feature ofm using the command **feature ofm**

Example:

```
switch (config)# feature ofm
```

Step 3 Enable static ERSPAN tunnels uisng the command **tunnel-profile** <profile-name>

Example:

```
switch(config)# tunnel-profile foo
                        switch(config-tnl-profile)#
```

Step 4 Set appropriate encapsulation type for the tunnel profile uisng the command **encapsulation erspan**

Example:

```
switch(config-tnl-profile)# encapsulation erspan
                        switch(config-tnl-profile)#
```

Step 5 Denote the monitored session that the related ERSPAN packet is associated with on the source switch using the command **erspan session-id all**

Example:

```
switch(config-tnl-profile)# erspan session-id all
```

Step 6 Add ethernet1/16 to the flow term list (if no flow CLI is configured) using the command **flow terminate interface add ethernet1/16**

Example:

```
switch(config-tnl-profile)# flow terminate interface add ethernet1/16
```

Step 7 Create an IPACL and enter the IP access list configuration mode using the command **ip access-list** <access-list-name>

Example:

```
switch(config)# ip access-list test
                        switch(config-acl)#
```

Step 8 Create an IP ACL rule that permits traffic to be redirected per its conditions using the command [**no**] **permit** protocol source destination **redirect** interfaces

Example:

```
permit ip any any redirect ethernet1/1,ethernet1/19
```

The no version of this command removes the permit rule from the policy.



Note When you enter an interface for the TAP aggregation policy, do not abbreviate it. When you enter a list of interfaces, separate them with commas, but no spaces.

Step 9 Apply the port access list to the ERSPAN strip/terminating port using the command **ip port access-group** <access-group name>_ **redir in**

Example:

```
interface e1/16 (config-if)# ip port access-group test in
```

The following example shows ERSPAN header stripping:

```
switch(config)# feature ofm
switch(config)# tunnel-profile foo
switch(config-tnl-profile)# encapsulation erspan
switch(config-tnl-profile)# erspan session-id all
switch(config-tnl-profile)# flowterminate interface add ethernet1/16
switch(config)# ip access-list test
permit ip any any redirect ethernet1/1,ethernet1/19
interfacee1/16 (config-if)# ip port access-group test in
```

What to do next

To display the ERSPAN header stripping configuration, perform one of the following tasks:

Command	Purpose
show run ofm	Displays the tunnel profiles.
show run aclmgr	Displays all the ACLs and the application of those ACLs on the interfaces.
show ip access-list acl_nam	Displays ACL hit and redirected packets count.
show tunnel-profile	Displays the states of all tunnel profiles.

GRE Header Stripping

Overview

The GRE header stripping feature removes the GRE header from encapsulated packets and adds a custom Ethernet header to the inner packet.

- After GRE strip, an Ethernet header is added to the inner packet.
- The inner packet in a GRE encapsulated packet does not contain an Ethernet header.
- The custom Ethernet header includes specific fields such as VLAN, destination MAC, and source MAC addresses.

Custom Ethernet Header Fields

After the GRE header is stripped, the following custom fields are added to the Ethernet header:

- 802.1q header with VLAN configured on the incoming port.
- Destination MAC address will be set to 00:00:00:ab:cd:ef or 000.000.abc.def.
- Source MAC address will be set to VDC MAC address of the switch.

The GRE header is removed for the following reasons :

- Some of the monitoring tools do not have the ability to understand an encapsulated packet.
- Addition of another header adds to the packet size which hampers the optimization of the amount of data that is sent to the tools.
- Presence of another header skews the analytics data.

Guidelines and Limitations

Platform Specifications

- A maximum of 500 flow terminate interfaces are supported on tunnel-profile with encap type: iVXLAN/VXLAN/GRE.
- A maximum of 31 flow terminate interfaces are supported on tunnel-profile with encap type: ERSPAN.

Not Supported Features

1. Flow terminate interfaces cannot share ESPRAN and GRE/VXLAN/IVXLAN profiles.
2. Feature OFM and feature tunnel cannot co-exist on the same switch.
3. The configuration of **mode tap-aggregation** should not be present on interfaces where GRE header stripping functionality is enabled.
4. Tunnel-encapsulation type modification is not allowed.

Configuration Guidelines

1. To remove a flow interface from a tunnel-profile, use **remove** instead of **no** . Using **no** in the flow terminate command deletes all interfaces from the flow terminate list.
2. If a GRE strip-enabled interface receives ERSPAN traffic, stripping succeeds, but traffic will not be forwarded to the redirect port.
3.
4. When flow terminate interface CLI is configured without **add** keyword, it acts as **replace** , which means previously added flow terminate interfaces are deleted and only new ones will act as flow terminate interfaces.
5. After non-disruptive upgrade from previous NX-OS version to 10.2(3)F, port ACL must be removed from all interfaces and added before enabling GRE header strip feature for a particular interface.
6. The **hardware acl tap-agg redirect disable-dot1q-sharing** command is required on 9300-GX to allow dot1q tunnel propagation. The switch needs reload after enabling this command.

Configuration

- Configuration for ingress ports includes setting the interface to access VLAN, enabling dot1q-tunnel mode, applying an IP access group, and enabling the interface.
- Configuration for egress ports includes setting the interface to trunk mode, enabling the interface, and applying an IP access list with specific permit and redirect rules.

Configuration Details for Egress and Ingress Ports

The following is the configuration for ingress ports:

- `interface eth1/1`

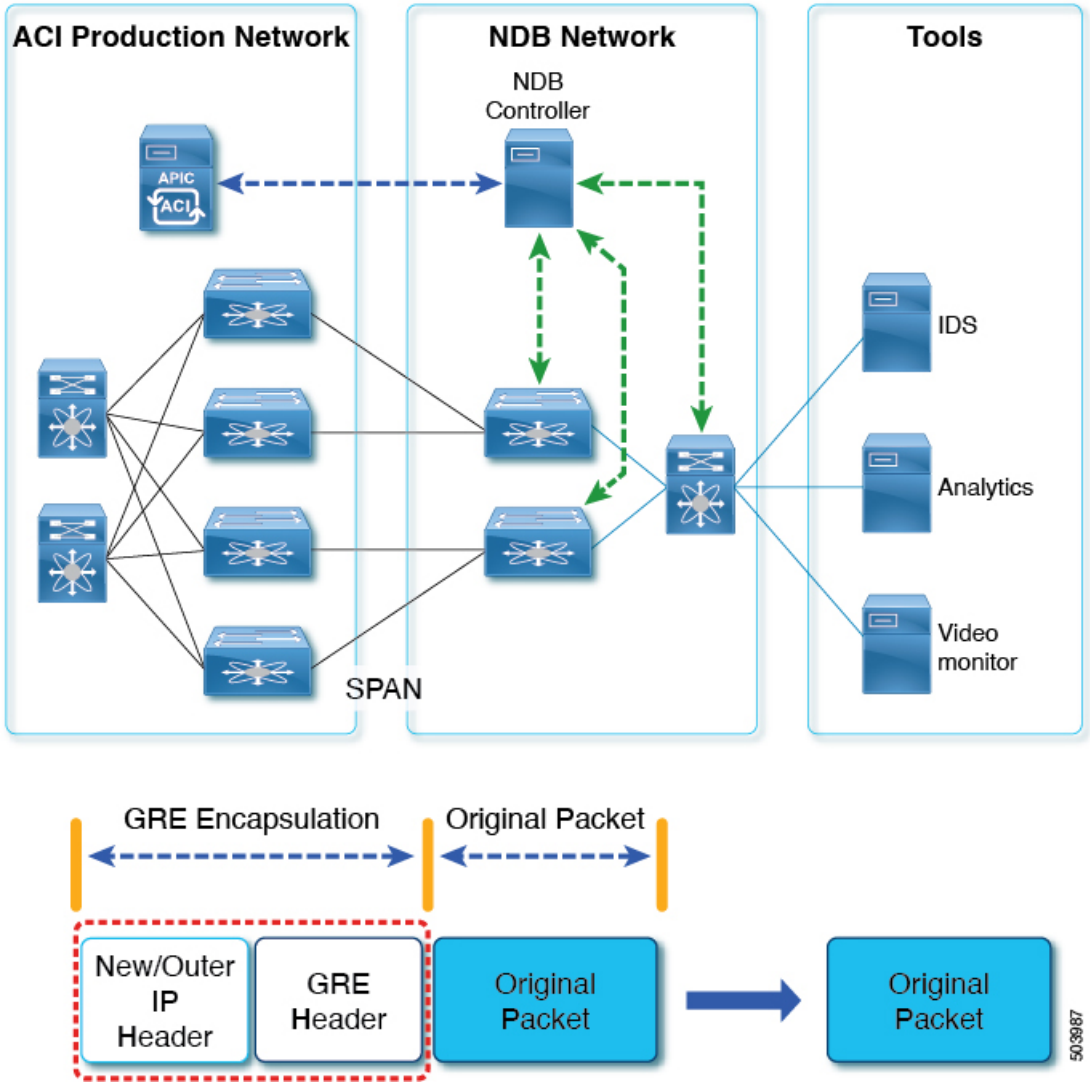
```
switchport access vlan 101
switchport mode dot1q-tunnel
ip port access-group ndb_acl in <<<
no shutdown
```

The following is the configuration for egress ports:

- `interface Ethernet1/7`

```
switchport mode trunk
no shutdown
IP access list ndb_acl
statistics per-entry
10 permit udp any any eq 4789 redirect Ethernet1/7
15 permit ip any any redirect Ethernet1/7
```

Figure 11: NDB GRE Header Strip Solution



CLIs for GRE Header Strip

The following are the CLIs to be configured for enabling GRE header on an interface:

```
feature ofm
tunnel-profile gre_strip
    encapsulation gre
    destination any
    flow terminate interface add Ethernet1/1-10
```

The following is the show command for tunnel-profile:

```
switch# show tunnel-profile gre_strip
  Profile               : gre_strip
  Encapsulation         : GRE
  State                 : UP
  Destination           : Any
  Terminate Interfaces  : 10
```

```
  Terminate List        : Ethernet1/1 Ethernet1/2 Ethernet1/3 Ethernet1/4 Ethernet1/5
Ethernet1/6 Ethernet1/7 Ethernet1/8 Ethernet1/9 Ethernet1/10
```

MPLS Header Stripping

Overview

MPLS label stripping is a process that removes the MPLS header from incoming packets and adds an Ethernet header with specific custom fields.

- It is supported for both IPoMPLS and EoMPLS packet formats.
- After label strip, an Ethernet header is added to the inner packet.
- The Ethernet header includes a VLAN tag, destination MAC, and source MAC address.

Custom Fields

After MPLS label stripping, the following custom fields are added to the Ethernet header:

- 802.1q header with VLAN configured on the incoming port.
- Destination MAC address is set to 00:00:00:ab:cd:ef or 000.000.abc.def.
- Source MAC address is set to VDC MAC address of the switch.

Guidelines and Limitations

Guidelines for Migration from legacy MPLS header stripping to OFM-based configuration

The following guidelines and limitations apply when migrating from legacy MPLS header stripping to OFM-based configuration:

Not Supported Features :

- Legacy MPLS stripping implementation cannot co-exist with any OFM-based stripping.
- Feature OFM and feature tunnel cannot co-exist on the same switch.
- Tunnel-encapsulation type modification is not allowed.

```
QP-CF-1(config-tnl-profile)# encapsulation mpls
Error: encap-type modify not allowed, delete and add again
```

- On an interface where MPLS head strip is enabled, mode tap-aggregation should not be present.
- MPLS Stripping is based on IP PACL, so do not use MAC-ACL for stripping.
- During MPLS stripping, incoming VLAN in the original packet is not preserved.
- After non-disruptive upgrade from previous NX-OS version to 10.2(3)F, port ACL must be removed from all interfaces and added before enabling MPLS header stripping feature for a particular interface.

- The **hardware acl tap-agg redirect disable-dot1q-sharing** command is required on Cisco Nexus 9300-GX platform switches to allow dot1q tunnel propagation. The switch needs reload after enabling this command.
- If ERSPAN ACL redirect tunnel-profile is not configured and the interface is receiving ERSPAN packets, then the ERSPAN packets will hit ERSPAN ACL redirect entries in TapAgg policy and will not be stripped.
- With ERSPAN tunnel-profile, when ingress interface is converted from dot1q-tunnel to trunk mode, egress packets will have dot1q tag with VLAN=1. This tagging takes place for both stripped packets and regular IP packets that are redirected.
- When an MPLS strip-enabled interface receives ERSPAN traffic, stripping succeeds, but traffic is not forwarded to the redirect port.
- To remove flow interface from a tunnel-profile, use **remove** instead of **no** . The use of **no** in flow terminate command will delete all interfaces from flow terminate list.

  For example:

```
switch(config)# tunnel-profile mpls_strip
switch(config-tnl-profile)# flow terminate interface remove Ethernet 1/48
```

- When flow terminate interface command is configured without the **add** keyword, it acts as **replace** , which means previously added flow terminate interfaces are deleted and only the new ones will act as flow terminate interfaces.
- MPLS packets with multicast bit set cannot be terminated, but MPLS packets with unicast bit set can be terminated.
- Ingress interface can be either in trunk mode or access mode. Both modes allow redirection of tagged and untagged packets. When access-mode is used along with dot1q-tunnel mode, after header stripping VLAN_tag is added as specified by the access-mode.

NDB MPLS Header Stripping Feature Guidelines

Beginning with Cisco NX-OS Release 10.2(3)F, the NDB MPLS Header Stripping feature is supported.

-

Note The OFM MPLS stripping feature is supported only on TORs; it is not supported on line cards.

Cleanup for Migrating from legacy MPLS stripping functionality

Migrating from legacy MPLS stripping functionality requires the following cleanup before enabling OFM-based MPLS stripping:

- Removal of **mode tap-aggregation** at interface(s) level
- Removal of **mpls strip; mpls strip dot1q** at the global level
- Save the configuration and reload the switch with the above configuration

EoMPLS Guidelines

- EoMPLS stripping can co-exist with all other header stripping features on same or different interfaces.
-
- Pseudo Wire Control Word is not supported.
- On Cisco Nexus 9300-GX platform switches, two ingress ports cannot share acl unless the dot1q vlan config is the same on them, else tagging does not work.

Configuration

- This topic provides configuration steps for ingress ports, including access VLAN assignment, dot1q-tunnel mode, and applying an IP access group.
- It also covers configuration for egress ports, such as trunk mode setup and applying IP access lists with redirect actions.
- These configurations are essential for managing traffic flow and security on network switches.

Configuration Details for Egress and Ingress Ports

The following sections provide the configuration commands for both ingress and egress ports on the switch.

- Ingress port configuration
- Egress port configuration

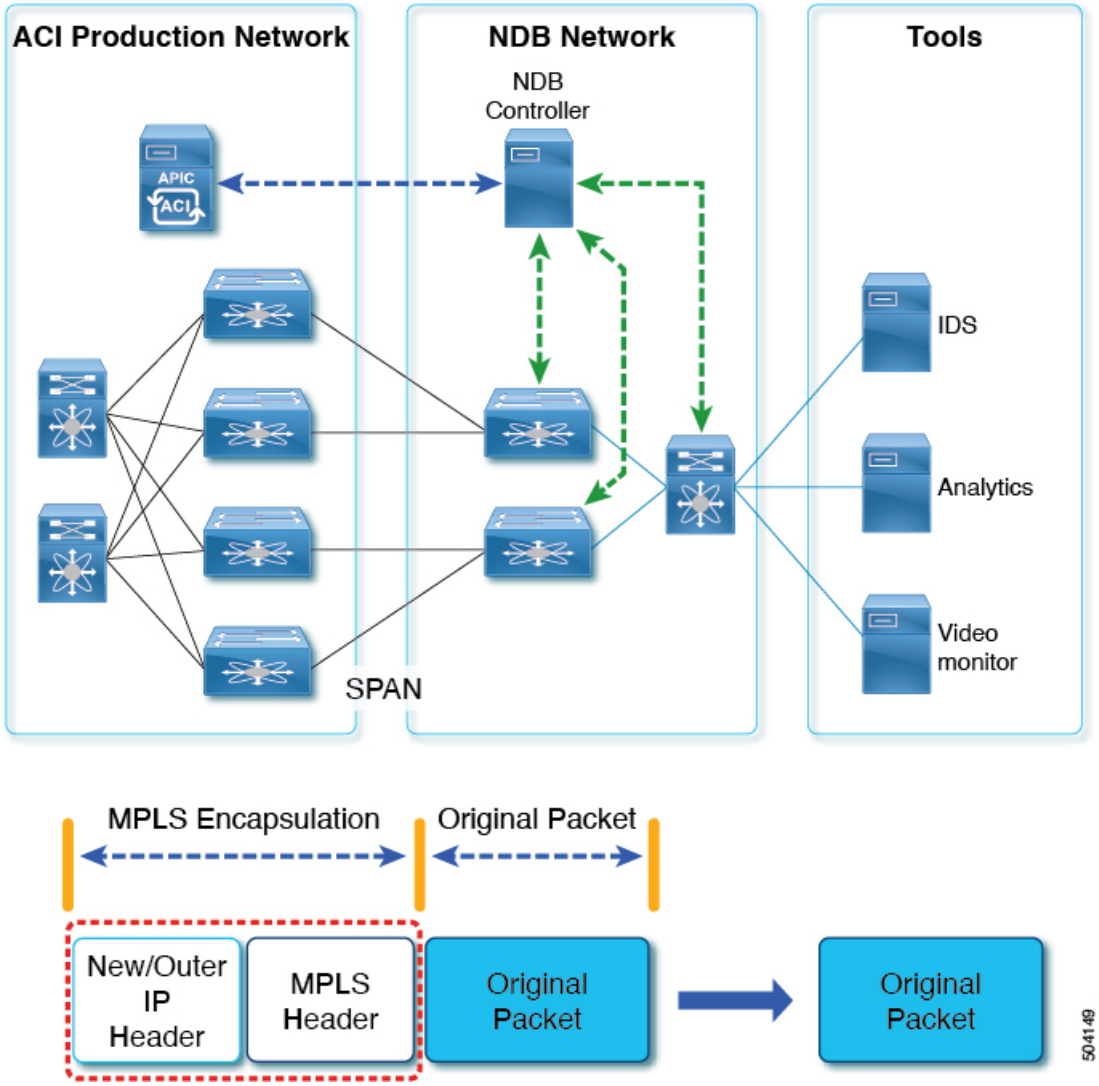
1. Configure the ingress port with access VLAN, dot1q-tunnel mode, and apply the IP access group.
2. Configure the egress port as a trunk and apply the IP access list with redirect actions.

- Ingress port: Assign VLAN, set mode, apply ACL
- Egress port: Set trunk mode, apply ACL with redirect



Note In case of decapsulated packet such as MPLS, the NDB-switch adds an Ethernet/VLAN header to the **original packet** , so egressing packet will have Ethernet/VLAN - original packet.

Figure 12: NDB MPLS Header Strip Solution



Example configuration for ingress port:

```
interface eth1/1
                  switchport access vlan 101
                  switchport mode dot1q-tunnel
                  ip port access-group ndb_acl in
                  no shutdown
```

Example configuration for egress port:

```
interface Ethernet1/7
                  switchport mode trunk
                  no shutdown
                  IP access list ndb_acl
                  statistics per-entry
                  10 permit udp any any eq 4789 redirect Ethernet1/7
                  15 permit ip any any redirect Ethernet1/7
```

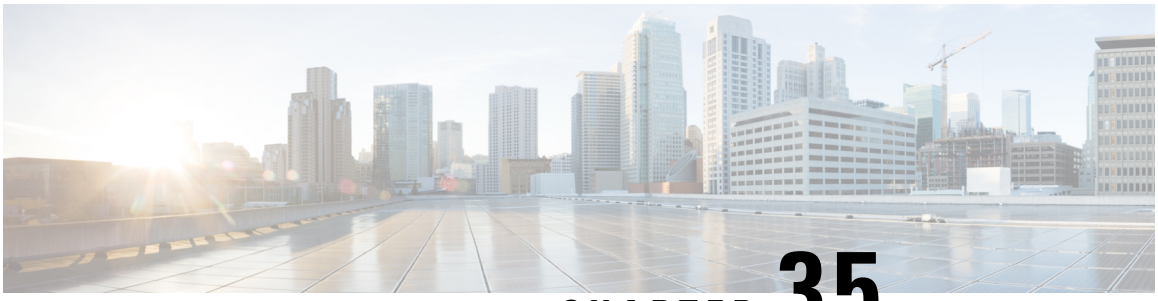
Commands for MPLS Header Strip Feature

The following commands should be configured for enabling MPLS header on an interface:

```
feature ofm
tunnel-profile
mpls_strip encapsulation mpls destination any
flow terminate interface add Ethernet1/1-10
```

The show command for tunnel-profile is as follows:

```
switch# show tunnel-profile mpls_strip
  Profile               : mpls_strip
  Encapsulation         : MPLS
  State                 : UP
  Destination           : Any
  Terminate Interfaces  : 10
  Terminate List        : Ethernet1/1 Ethernet1/2 Ethernet1/3 Ethernet1/4 Ethernet1/5
Ethernet1/6 Ethernet1/7 Ethernet1/8 Ethernet1/9 Ethernet1/10
```



CHAPTER 35

Graceful Insertion and Removal

- Overview, on page 565
- Guidelines and Limitations, on page 567
- Configure Maintenance-Mode Profile, on page 568
- Configure Normal-Mode Profile, on page 571
- Create Snapshot, on page 572
- Add Show Commands to Snapshots, on page 574
- Trigger Graceful Removal, on page 575
- Trigger Graceful Insertion, on page 578
- Maintenance Mode Enhancements, on page 579
- Verify Configuration, on page 580
- Configuration Examples, on page 581

Overview

You can use graceful insertion and removal to gracefully eject a switch and isolate it from the network in order to perform debugging or upgrade operations. The switch is removed from the regular forwarding path with minimal traffic disruption. When you are finished performing debugging or upgrade operations, you can use graceful insertion to return the switch to its fully operational (normal) mode.

When you place the switch in maintenance mode, all configured Layer 3 control-plane protocols are isolated from the network. Directly connected routes are not withdrawn or modified during this state. When normal mode is restored, the advertisement of all routes is restored.

In graceful removal, all protocols and vPC domains are gracefully brought down and the switch is isolated from the network. In graceful insertion, all protocols and vPC domains are restored.

The following protocols are supported (for both IPv4 and IPv6 address families):

- Border Gateway Protocol (BGP)
- Enhanced Interior Gateway Routing Protocol (EIGRP)
- Intermediate System-to-Intermediate System (ISIS)
- Open Shortest Path First (OSPF)
- Protocol Independent Multicast (PIM)
- Routing Information Protocol (RIP)

Type of Profiles

Note For graceful insertion and removal, the PIM protocol is applicable only to vPC environments. During graceful removal, the vPC forwarding role is transferred to the vPC peer for all northbound sources of multicast traffic.

By default, the system isolates all enabled protocols during graceful removal and restores them during graceful insertion. The protocols are isolated and restored in a predefined order.

If you want to isolate, shut down, or restore the protocols individually (or perform additional configurations), you can create a profile with configuration commands that can be applied during graceful removal or graceful insertion. However, you need to make sure that the order of the protocols is correct and any dependencies are considered.

The switch supports the following profiles:

- Maintenance-mode profile—Contains all the commands that will be executed during graceful removal, when the switch enters maintenance mode.
- Normal-mode profile—Contains all the commands that will be executed during graceful insertion, when the switch returns to normal mode.

The following commands (along with any configuration commands) are supported in the profiles.

Note The **shutdown** command takes precedence when both **shutdown** and **isolate** are configured under a routing protocol instance or maintenance-mode profile.

Command	Description
isolate	Isolates the protocol from the switch and puts the protocol in maintenance mode.
no isolate	Restores the protocol and puts the protocol in normal mode.
shutdown	Shuts down the protocol or vPC domain.
no shutdown	Brings up the protocol or vPC domain.
system interface shutdown [**exclude fex-fabric**]	Shuts down the system interfaces (except the management interface).
no system interface shutdown [**exclude fex-fabric**]	Brings up the system interfaces.
sleep instance *instance-number seconds*	Delays the execution of the command by a specified number of seconds. You can delay multiple instances of the command. The range for the *instance-number* and *seconds* arguments is from 0 to 2177483647.

Command	Description
python instance *instance-number uri* [*python-arguments*] Example: **python instance 1 bootflash://script1.py**	Configures Python script invocations to the profile. You can add multiple invocations of the command to the profile. You can enter a maximum of 32 alphanumeric characters for the Python arguments.



Note Beginning with Cisco NX-OS Release 9.3(5), the **isolate** command is provided with the **include-local** option, which is applicable only to **router bgp**.

If you use this option, BGP withdraws all the routes from its peers. If you do not use this option, then BGP only withdraws remotely learned routes, and the locally originated routes such as aggregate, injected, network and redistribute continue to be advertised with maximum Multi-Exit Discriminator (MED) to eBGP peers and minimum local preference to iBGP peers.

Workflow

Follow these steps to complete the graceful insertion and removal (GIR) workflow:

1. (Optional) Create the maintenance-mode profile. (See Configure Maintenance-Mode Profile, on page 568.)
2. (Optional) Create the normal-mode profile. (See Configure Normal-Mode Profile, on page 571.)
3. Take a snapshot before triggering graceful removal. (See Create Snapshot, on page 572.)
4. Trigger graceful removal to put the switch in maintenance mode. (See Trigger Graceful Removal, on page 575.)
5. Trigger graceful insertion to return the switch to normal mode. (See Trigger Graceful Insertion, on page 578.)
6. Take a snapshot after triggering graceful insertion. (See Create Snapshot, on page 572.)
7. Use the **show snapshots compare** command to compare the operational data before and after the graceful removal and insertion of the switch to make sure that everything is running as expected. (See Verify Configuration, on page 580.)

Guidelines and Limitations

General Guidelines

- Remove all existing custom profiles before creating new custom profiles for normal-mode and maintenance-mode.

Release Specific Guidelines

- Beginning with Cisco NX-OS Release 9.2(1), L2 Graceful Insertion and Replacement is supported. When moving from normal to maintenance mode, MCT goes down resulting in north to south traffic convergence. Zero packet loss is not supported. The following table provides an example of traffic convergence of 10 vPCs with 2 port member on each VPC port and 60k mac scale.

  Table 41:

Trigger	Role	North to South Traffic	South to North Traffic
Normal to maintenance mode	Primary	760 ms	1320 ms
Maintenance mode to normal	Primary	13155 ms	27980 ms
Normal to maintenance mode	Secondary	300 ms	1375 ms
Maintenance mode to normal	Secondary	15905 ms	23350 ms

- Beginning with Cisco NX-OS Release 9.2(1), if you configure the isolate option for OSPF, direct routes and stub routes are advertised as max-metric routes. As a result, north-to-south traffic to the SVI hosts goes through the vPC peer when only one vPC switch is isolated.
- Beginning with Cisco NX-OS Release 9.3(5), the **include-local** option is added to the existing **isolate** command. However, the **include-local** option applies only to **router bgp**.

  The default maintenance-mode profile can cause an Anycast BGW to drop multi-fabric BUM traffic for VNIs on which the BGW is the designated forwarder. Use a custom profile that configures `isolate include-local` under BGP when the BGW enters maintenance mode. Configure `no isolate include-local` under BGP in the corresponding normal-mode profile.
- Beginning with Cisco NX-OS Release 10.3(1)F, during a binary reload scenario, when the system is switched to maintenance mode, the interface is shut until the system boots up completely and declares system readiness.

N9300 Smart Switches Guidelines

- Beginning with NX-OS Release 10.6(2)F, Graceful Insertion and Replacement (GIR) is supported on the N9324C-SE1U and N9348Y2C6D-SE1U switches.

Configure Maintenance-Mode Profile

You can create a maintenance-mode profile with configuration commands that can be applied during graceful removal or graceful insertion.

Note During the maintenance mode the SVI becomes UP after the reload. In this scenario, use the **isolate include-local** command under router BGP or keep interfaces in shutdown state through maintenance mode to avoid the impact of advertising connected/static routes.

Procedure

Step 1 Enter a configuration session for the maintenance-mode profile using the command [**no**] **configure maintenance profile maintenance-mode**

Example:

```
switch# configure maintenance profile maintenance-mode
      Enter configuration commands, one per line. End with CNTL/Z.
      switch(config-mm-profile)#
```

The **no** option deletes the maintenance profile maintenance-mode.

Depending on which protocols you have configured, you must now enter the appropriate commands to bring down the protocols. For a list of supported commands, see Profiles .

Step 2 Close the maintenance-mode profile using the command **end**

Example:

```
switch(config-mm-profile)# end
 switch#
```

Step 3 Display the details of the maintenance-mode profile using the command **show maintenance profile maintenance-mode**

Example:

```
switch# show maintenance profile maintenance-mode
```

This example shows how to create a maintenance-mode profile:

```
switch#
configure maintenance profile maintenance-mode
Enter configuration commands, one per line. End with CNTL/Z.
switch(config-mm-profile)#
ip pim isolate
switch(config-mm-profile)#
router bgp 100
switch(config-mm-profile-router)#
shutdown
switch(config-mm-profile)#
router eigrp 10
switch(config-mm-profile-router)#
shutdown
switch(config-mm-profile-router)#
address-family ipv6 unicast
switch(config-mm-profile-router-af)#
shutdown
```

```
switch(config-mm-profile)#
vpc domain 10
switch(config-mm-profile-config-vpc-domain)#
shutdown
switch(config-mm-profile)#
system interface shutdown
switch(config-mm-profile)#
end
Exit maintenance profile mode.
switch#
show maintenance profile maintenance-mode
[Maintenance Mode]
ip pim isolate
router bgp 100
shutdown
router eigrp 10
shutdown
address-family ipv6 unicast
shutdown
vpc domain 10
shutdown
system interface shutdown
```

This example shows how to configure sleep instance in a custom profile to add a delay before the next protocol change.

```
switch#
configure maintenance profile maintenance-mode
Enter configuration commands, one per line. End with CNTL/Z.
switch(config-mm-profile)#
router bgp 65001
switch(config-mm-profile-router)#
isolate
switch(config-mm-profile-router)#
sleep instance 1 10
switch(config-mm-profile)#
router eigrp 200
switch(config-mm-profile-router)#
isolate
switch(config-mm-profile-router)#
sleep instance 2 15
switch(config-mm-profile)#
router ospf 100
switch(config-mm-profile-router)#
isolate
switch(config-mm-profile-router)#
sleep instance 3 20
switch(config-mm-profile)#
router ospfv3 300
switch(config-mm-profile-router)#
isolate
switch(config-mm-profile-router)#
sleep instance 4 5
switch(config-mm-profile)#
router isis 400
switch(config-mm-profile-router)#
isolate
switch(config-mm-profile)#
end
Exit maintenance profile mode.
switch#
```

Note If you need to run exec commands or add a dynamic delay while the maintenance mode profile is applied, use the **python instance** *instance-number uri* [*python-arguments*] script.

Configure Normal-Mode Profile

You can create a normal-mode profile with configuration commands that can be applied during graceful removal or graceful insertion.

Procedure

Step 1 Enter a configuration session for the normal-mode profile using the command [**no**] **configure maintenance profile normal-mode**

Example:

```
switch# configure maintenance profile normal-mode
      Enter configuration commands, one per line. End with CNTL/Z.
      switch(config-mm-profile)#
```

The **no** version removes the maintenance profile normal-mode.

Depending on which protocols you have configured, you must now enter the appropriate commands to bring up the protocols. For a list of supported commands, see Profiles .

Step 2 Close the normal-mode profile using the command **end**

Example:

```
switch(config-mm-profile)# end
 switch#
```

Step 3 Display the details of the normal-mode profile using the command **show maintenance profile normal-mode**

Example:

```
switch# show maintenance profile normal-mode
```

This example shows how to create a maintenance profile normal-mode:

```
switch#
configure maintenance profile normal-mode
switch(config-mm-profile)#
no system interface shutdown
switch(config-mm-profile)#
vpc domain 10
switch(config-mm-profile-config-vpc-domain)#
no shutdown
switch(config-mm-profile)#
router eigrp 10
switch(config-mm-profile-router)#
```

```
no shutdown
switch(config-mm-profile-router)#
address-family ipv6 unicast
switch(config-mm-profile-router-af)#
no shutdown
switch(config-mm-profile)#
router bgp 100
switch(config-mm-profile-router)#
no shutdown
switch(config-mm-profile)#
no ip pim isolate
switch(config-mm-profile)#
end
Exit maintenance profile mode.
switch#
show maintenance profile normal-mode
[Normal Mode]
no system interface shutdown
vpc domain 10
no shutdown
router eigrp 10
no shutdown
address-family ipv6 unicast
no shutdown
router bgp 100
no shutdown
no ip pim isolate
```

Create Snapshot

You can create a snapshot of the running states of selected features. When you create a snapshot, a predefined set of **show** commands are run and the outputs are saved.

Procedure

Step 1 Capture the running state or operational data of selected features and stores the data on persistent storage media using the command **snapshot create** *snapshot-name description*

Example:

```
switch# snapshot create snap_before_maintenance Taken before maintenance
 Executing 'show interface'... Done
 Executing 'show ip route summary vrf all'... Done
 Executing 'show ipv6 route summary vrf all'... Done
 Executing 'show bgp sessions vrf all'... Done
 Executing 'show ip eigrp topology summary'... Done
 Executing 'show ipv6 eigrp topology summary'... Done
 Feature 'vpc' not enabled, skipping...
 Executing 'show ip ospf vrf all'... Done
 Feature 'ospfv3' not enabled, skipping...
 Feature 'isis' not enabled, skipping...
 Feature 'rip' not enabled, skipping...
 Snapshot 'snap_before_maintenance' created
```

You can enter a maximum of 64 alphanumeric chapters for the snapshot name and a maximum of 254 alphanumeric characters for the description.

Use the **snapshot delete** { **all** | *snapshot-name* } command to delete all snapshots or a specific snapshot.

Step 2 Display snapshots present on the switch using the command **show snapshots**

Example:

```
switch# show snapshots
 Snapshot Name             Time                         Description
 ---------------------------------------------------------------------------
 snap_before_maintenance   Wed Aug 19 13:53:28 2015  Taken before maintenance
```

Step 3 Display a comparison of two snapshots using the command **show snapshots compare** *snapshot-name-1 snapshot-name-2* [**summary** | **ipv4routes** | **ipv6routes**]

Example:

```
switch# show snapshots compare snap_before_maintenance snap_after_maintenance
```

The **summary** option displays just enough information to see the overall changes between the two snapshots.

The **ipv4routes** and **ipv6routes** options display the changes in IPv4 and IPv6 routes between the two snapshots.

The following example shows a summary of the changes between two snapshots:

```
switch#
show snapshots compare snapshot1 snapshot2 summary
feature                              snapshot1      snapshot2     changed
basic summary
# of interfaces                 16             12            *
# of vlans                      10             4             *
# of ipv4 routes                33             3             *
……
interfaces
# of eth interfaces             3              0             *
# of eth interfaces up          2              0             *
# of eth interfaces down        1              0             *
# of eth interfaces other       0              0
# of vlan interfaces            3              1             *
# of vlan interfaces up         3              1             *
# of vlan interfaces down       0              0
# of vlan interfaces other      0              1             *
……
```

The following example shows the changes in IPv4 routes between two snapshots:

```
switch#
show snapshots compare snapshot1 snapshot2 ipv4routes
metric                          snapshot1      snapshot2       changed
# of routes                     33             3               *
# of adjacencies                10             4               *
Prefix             Changed Attribute
------             -----------------
23.0.0.0/8         not in snapshot2
10.10.10.1/32      not in snapshot2
21.1.2.3/8         adjacency index has changed from 29 (snapshot1) to 38 (snapshot2)
……
```

```
There were 28 attribute changes detected
```

Add Show Commands to Snapshots

You can specify additional **show** commands to be captured in snapshots. These **show** commands are defined in user-specified snapshot sections.

Procedure

Step 1 Add a user-specified section to snapshots using the command **snapshot section add** *section "show-command" row-id element-key1* [*element-key2*]

Example:

```
switch# snapshot section add myshow "show ip interface brief" ROW_intf intf-name
```

The *section* is used to name the **show** command output. You can use any word to name the section.

The **show** command must be enclosed in quotation marks. Non- **show** commands will not be accepted.

The *row-id* argument specifies the tag of each row entry of the **show** command's XML output. The *element-key1* and *element-key2* arguments specify the tags used to distinguish among row entries. In most cases, only the *element-key1* argument needs to specified to be able to distinguish among row entries.

Note To delete a user-specified section from snapshots, use the **snapshot section delete** *section* command.

Step 2 Display the user-specified snapshot sections using the command **show snapshots sections**

Example:

```
switch# show snapshots sections
```

The following example adds the **show ip interface brief** command to the myshow snapshot section.

```
switch#
snapshot section add myshow "show ip interface brief" ROW_intf intf-name
switch#
show snapshots sections
user-specified snapshot sections
--------------------------------
[myshow]
cmd:  show ip interface brief
row:  ROW_intf
key1: intf-name
key2: -
[sect2]
cmd:  show ip ospf vrf all
row:  ROW_ctx
key1: instance_number
```

```
key2: cname
```

Trigger Graceful Removal

Use this task to trigger a graceful removal of the switch, which will eject the switch and isolate it from the network for debugging or upgrade operations.

In order to perform debugging or upgrade operations, you can trigger a graceful removal of the switch, which will eject the switch and isolate it from the network.

Before you begin

If you want the system to use a maintenance-mode profile that you create, see Configure Maintenance-Mode Profile, on page 568 .

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Put all enabled protocols in maintenance mode (using the **isolate** command) **system mode maintenance** [**dont-generate-profile** | **timeout** *value* | **shutdown** | **on-reload reset-reason** *reason*]

Example:

```
switch(config)# system mode maintenance
Following configuration will be applied:
ip pim isolate
router bgp 65502
isolate
router ospf p1
isolate
router ospfv3 p1
isolate
Do you want to continue (y/n)? [no]
y
Generating a snapshot before going into maintenance mode
Starting to apply commands...
Applying : ip pim isolate
Applying : router bgp 65502
Applying :   isolate
Applying : router ospf p1
Applying :   isolate
Applying : router ospfv3 p1
Applying :   isolate
Maintenance mode operation successful.
```

The following options are available:

- **dont-generate-profile** —Prevents the dynamic searching of enabled protocols and executes commands configured in a maintenance-mode profile. Use this option if you want the system to use a maintenance-mode profile that you have created.
- **timeout** *value* —Keeps the switch in maintenance mode for a specified number of minutes. The range is from 5 to 65535. Once the configured time elapses, the switch returns to normal mode automatically. The **no system mode maintenance timeout** command disables the timer.



Note The timeout value 65535 is the default setting and is treated as "no timeout." This value will not appear in the running configuration/show running-config all.

- **shutdown** —Shuts down all protocols, vPC domains, and interfaces except the management interface (using the **shutdown** command). This option is disruptive while the default (which uses the **isolate** command) is not.
- **on-reload reset-reason** *reason* —Boots the switch into maintenance mode automatically in the event of a specified system crash. The **no system mode maintenance on-reload reset-reason** command prevents the switch from being brought up in maintenance mode in the event of a system crash.

  The maintenance mode reset reasons are as follows:

  - HW_ERROR—Hardware error
  - SVC_FAILURE—Critical service failure
  - KERN_FAILURE—Kernel panic
  - WDOG_TIMEOUT—Watchdog timeout
  - FATAL_ERROR—Fatal error
  - LC_FAILURE—Line card failure
  - MATCH_ANY—Any of the above reasons



Note

- During reload, a binary form of configuration file is used to restore the system. However, this is not possible in all scenarios, for example, when the reloaded image is new. Then, the system switches to ASCII reload.
- During an unexpected reload, when the system switches to maintenance mode, the interface will be in a shut state (reason: **mmodeBootIntfShut**), until the system comes up completely from normal mode to maintenance mode.
- During reload, if the system does not come up completely for a duration that is longer than expected, then collect the debugging information using the following commands and contact Cisco Support :
  - **show tech support mmode**
  - **show tech support system manager**
  - **show tech support interface manager**
  - **show accounting**

The system prompts you to continue. Enter **y** to continue or **n** to terminate the process.

Step 3 Display the current system mode using the command **show system mode**

Example:

```
switch(config)# show system mode
 System Mode: Maintenance
```

This is

This is an optional step. The switch is in maintenance mode. You can now perform any desired debugging or upgrade operations on the switch.

Step 4 Copy the running configuration to the startup configuration using the command **copy running-config startup-config**

Example:

```
switch(config)# copy running-config startup-config
```

This is an optional step. This command is required if you want to preserve maintenance mode following a reboot.

This example shows how to shut down all protocols, vPC domains, and interfaces on the switch:

```
switch(config)#
system mode maintenance shutdown
Following configuration will be applied:
vpc domain 10
shutdown
router bgp 65502
shutdown
router ospf p1
shutdown
router ospfv3 p1
shutdown
system interface shutdown
Do you want to continue (y/n)? [no]
y
Generating a snapshot before going into maintenance mode
Starting to apply commands...
Applying : vpc domain 10
Applying :   shutdown
Applying : router bgp 65502
Applying :   shutdown
Applying : router ospf p1
Applying :   shutdown
Applying : router ospfv3 p1
Applying :   shutdown
Maintenance mode operation successful.
```

This example shows how to automatically boot the switch into maintenance mode if a fatal error occurs:

```
switch(config)#
   system mode maintenance on-reload reset-reason fatal_error
```

Trigger Graceful Insertion

When you finish performing any debugging or upgrade operations, you can trigger a graceful insertion to restore all protocols.

Before you begin

If you want the system to use a normal-mode profile that you create, see Configure Maintenance-Mode Profile, on page 568 .

Procedure

Step 1 Enter global configuration mode using the command **configure terminal**

Example:

```
switch# configure terminal
 switch(config)#
```

Step 2 Put all enabled protocols in normal mode (using the **no isolate** command) **no system mode maintenance** [**dont-generate-profile**]

Example:

```
switch(config)# no system mode maintenance dont-generate-profile
Following configuration will be applied:
no ip pim isolate
router bgp 65502
no isolate
router ospf p1
no isolate
router ospfv3 p1
no isolate
Do you want to continue (y/n)? [no]
     y
Starting to apply commands...
Applying : no ip pim isolate
Applying : router bgp 65502
Applying :   no isolate
Applying : router ospf p1
Applying :   no isolate
Applying : router ospfv3 p1
      Applying :   no isolate
Maintenance mode operation successful.
Generating Current Snapshot
```

The **dont-generate-profile** option prevents the dynamic searching of enabled protocols and executes commands configured in a normal-mode profile. Use this option if you want the system to use a normal-mode profile that you have created.

The system prompts you to continue. Enter **y** to continue or **n** to terminate the process.

Step 3 (Optional) Display the current system mode using the command **show system mode**

Example:

```
switch(config)# show system mode
 System Mode: Normal
```

This is an optional command. The switch is now in normal mode and is fully operational.

Maintenance Mode Enhancements

Starting with Release 7.0(3)I5(1), the following maintenance mode enhancements have been added to Cisco Nexus 9000 Series switches:

- In the system maintenance shutdown mode, the following message is added:

  ```
  NOTE: The command system interface shutdown will shutdown all interfaces excluding mgmt
   0.
  ```

- Entering the CLI command, **system mode maintenance** checks and sends alerts for the orphan ports.
- In isolate mode, when the vPC is configured, the following message is added:

  ```
  NOTE: If you have vPC orphan interfaces, please ensure vpc orphan-port suspend is
  configured under them, before proceeding further.
  ```

- Custom Profile Configuration: A new CLI command, **system mode maintenance always-use-custom-profile** is added for custom profile configuration. A new CLI command, **system mode maintenance non-interactive** is added for Cisco Nexus 9000 Series switches only. It provides a way to facilitate the transition to maintenance mode or normal mode without confirmation being done or each step being printed on the CLI session.

  When a loopback interface is configured with an IP address on a device, and this device is advertised to a peer device, then the device (with the loopback interface) moves to maintenance mode. In such a case, use the custom maintenance profile when **system interface shutdown** is configured on the device.

  When you create a custom profile (in maintenance or normal mode), it displays the following message:

  ```
  Please use the command

   system mode maintenance always-use-custom-profile

   if you want to always use the custom profile.
  ```

- A delay has been added before the after_maintenance snapshot is taken. The **no system mode maintenance** command exits once all the configuration for the normal mode has been applied, the mode has been changed to normal mode, and a timer has been started to take the after_maintenance snapshot. Once the timer expires, the after_maintenance snapshot is taken in the background and a new warning syslog, MODE_SNAPSHOT_DONE is sent once the snapshot is complete.

  The final output of the CLI command **no system mode maintenance** indicates when the after_maintenance snapshot is generated:

  ```
  The after_maintenance snapshot will be generated in <delay> seconds.
  After that time, please use show snapshots compare before_maintenance
  after_maintenance to check the health of the system. The timer delay
  ```

`for the after_maintenance snapshot is defaulted to 120 seconds but it can be changed by a new configuration command.`

The new configuration command to change the timer delay for the after_maintenance snapshot is **system mode maintenance snapshot-delay <seconds>** . This configuration overrides the default setting of 120 seconds to any value between 0 and 65535 and it is displayed in the ASCII configuration.

A new show command, **show maintenance snapshot-delay** has also been added to display the current snapshot-delay value. This new show command supports the XML output.

- A visible CLI indicator has been added to display when the system is in the maintenance mode, for example, `switch(maint-mode)#` .
- Support for the SNMP traps has been added when the device moves from the maintenance mode to the normal mode and vice-versa through CLI reload, or system reset. The **snmp-server enable traps mmode cseMaintModeChangeNotify** trap is added to enable changing to the maintenance mode trap notification. The **snmp-server enable traps mmode cseNormalModeChangeNotify** is added to enable changing to the normal mode trap notification. Both the traps are disabled by default.
- During maintenance mode, you can remove router from forward traffic with BGP active. You can advertise route installed locally to its iBGP peers with local setting in the absence of graceful shutdown. From Cisco NX-OS Release 10.4(1)F, you can configure custom isolation on route-map. In this mode, you can configure a route-map to choose the routes for BGP to be advertise.

Verify Configuration

To display the GIR configuration, perform one of the following tasks:

Command	Purpose
show interface brief	Displays abbreviated interface information.
show maintenance on-reload reset-reasons	Displays the reset reasons for which the switch comes up in maintenance mode. For a description of the maintenance mode reset reasons, see Trigger Graceful Removal, on page 575 .
show maintenance profile [**maintenance-mode** | **normal-mode**]	Displays the details of the maintenance-mode or normal-mode profile.
show maintenance timeout	Displays the maintenance-mode timeout period, after which the switch automatically returns to normal mode.
show { **running-config** | **startup-config** } **mmode** [**all**]	Displays the maintenance-mode section of the running or startup configuration. The **all** option includes the default values.
show snapshots	Displays snapshots present on the switch.

Command	Purpose
show snapshots compare *snapshot-name-1 snapshot-name-2* [**summary** | **ipv4routes** | **ipv6routes**]	Displays a comparison of two snapshots. The **summary** option displays just enough information to see the overall changes between the two snapshots. The **ipv4routes** and **ipv6routes** options display the changes in IPv4 and IPv6 routes between the two snapshots.
show snapshots dump *snapshot-name*	Displays the content of each file that was generated when the snapshot was taken.
show snapshots sections	Displays the user-specified snapshot sections.
show system mode	Displays the current system mode.

Configuration Examples

The **redistribute direct** configuration under Border Gateway Protocol (BGP) will attract traffic as the BGP isolate mode does not withdraw direct routes. This example shows how to use the **route-map** command to enable BGP to withdraw direct routes in isolate mode.

Policy Configuration

From Cisco NX-OS Release 10.4(1)F, you can match locally originated, redistributed direct routes.Configure route maps using below command:

Configuring redistribute for route-map

```
switch(config)# route-map <route map name>
switch(config-routemap)# match route-type redistributed-direct
```

Use the **route-map my-rmap-deny** command in maintenance mode to exclude SVIs with a tag 200 configuration.

```
switch(config)# route-map my-rmap-deny deny 10
switch(config-route-map)# match tag 200
switch(config-route-map)# exit
switch(config)# route-map my-rmap-deny permit 20
```

Use the **route-map my-rmap-permit** command in normal mode to include SVIs with a tag 200 configuration.

```
switch(config)# route-map my-rmap-permit permit 10
switch(config-route-map)# match tag 200
switch(config-route-map)# exit
switch(config)# route-map my-rmap-permit permit 20
```

Virtual IP (vIP)/Switch Virtual Interface (SVI) Configuration

```
switch(config)# interface loopback 200
switch(config-if)# ip address 192.0.2.100/8 tag 200
switch(config)# interface vlan 2
switch(config-if)# ip address 192.0.2.108/8 tag 200
....
switch(config)# interface vlan 3
switch(config-if)# ip address 192.0.2.102/8 tag 200
```

BGP Configuration

```
switch(config)# feature bgp
switch(config)# router bgp 100
switch(config-router)# neighbor 192.0.2.100
....
```

Maintenance Mode Profile

```
switch# configure maintenance profile maintenance-mode
switch(config-mm-profile)# router bgp 200
switch(config-mm-profile-router)# address-family ipv4 unicast
switch(config-mm-profile-router-af)# redistribute direct route-map my-rmap-deny
switch(config-mm-profile-router-af)# exit
switch(config-mm-profile)# sleep instance 1 10
```

Normal Mode Profile

```
switch# configure maintenance profile normal-mode
switch(config-mm-profile)# router bgp 100
switch(config-mm-profile-router)# address-family ipv4 unicast
switch(config-mm-profile-router-af)# redistribute direct route-map my-rmap-permit
switch(config-mm-profile-router-af)# exit
switch(config-mm-profile)# sleep instance 1 20
```

Custom Isolation Mode Profile

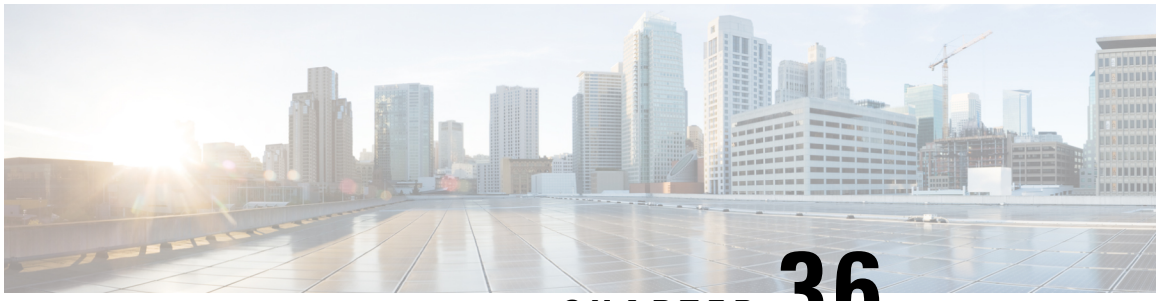
```
switch# configure maintenance profile maintenance-mode
switch(config-mm-profile)# router bgp 100
switch(config-mm-profile-router)# isolate route-map <route map name>
```

Anycast BGW Normal Mode Profile

```
configure maintenance profile normal-mode
no ip pim isolate
router bgp 65001
no isolate include-local
```

Anycast BGW Maintenance Mode Profile

```
configure maintenance profile maintenance-mode
ip pim isolate
router bgp 65001
isolate include-local
```



CHAPTER 36

Software Maintenance Upgrades

- Overview, on page 583
- Guidelines and Limitations, on page 585
- Prepare for Package Installation, on page 586
- Download SMU Package File from Cisco.com, on page 588
- Copy Package File to Local Storage Device or Network Server, on page 588
- Add and Activate Package, on page 590
- Commit Active Package Set, on page 592
- Deactivate and Remove Packages, on page 593
- No-Reload Options for SMU Installation, on page 595
- Advanced SMU Installation Methods, on page 600
- Downgrade Feature RPMs, on page 603
- Display Installation Log Information, on page 605
- Perform Software Maintenance Upgrade for Guest Shell Bash, on page 605
- Related Documents, on page 606
- History, on page 607

Overview

A software maintenance upgrade (SMU) is a package file that contains fixes for a specific defect. SMUs are created to respond to immediate issues and do not include new features. Typically, SMUs do not have a large impact on device operations. SMU versions are synchronized to the package major, minor, and maintenance versions they upgrade.

The effect of an SMU depends on its type:

- Process restart SMU-Causes a process or group of processes to restart on activation.
- Reload SMU-Causes a parallel reload of supervisors and line cards.

SMUs are not an alternative to maintenance releases. They provide a quick resolution of critical issues. All defects fixed by SMUs are integrated into the next maintenance releases of upcoming software trains, as applicable. SMUs also have the following considerations:

- SMUs are created for the following:
    - Critical SIR PSIRTs without a workaround or fix

- Severity1 and Severity2 issues without a workaround or fix

- If a fix is already available in a maintenance release of the same software train or already released on a later long-lived release, no SMU is provided. You are encouraged to acquire the fix from the maintenance release.

Note Depending on the fix, in some cases it may not be possible to provide an SMU. In such cases, the only option is to upgrade to the next maintenance release when available.

For information on upgrading your device to a new feature or maintenance release, see the Cisco Nexus 9000 Series NX-OS Software Upgrade and Downgrade Guide .

For information on Cisco NX-OS optionality feature, see the Cisco Nexus 9000 Series NX-OS Software Upgrade and Downgrade Guide .

Note Activating an SMU does not cause any earlier SMUs, or the package to which the SMU applies, to be automatically deactivated.

Impact of Package Activation and Deactivation

The activation or deactivation of an SMU package can have an immediate impact on the system. The system can be affected in the following ways:

- New processes might be started.
- Running processes might be stopped or restarted.
- All processes in the line cards might be restarted. Restarting processes in the line cards is equivalent to a soft reset.
- The line cards might reload.
- No processes in the line cards might be affected.

Note You must address any issues that result from the revised configuration and reapply the configuration, if necessary.

Tip After the activation process completes, enter the **show install log** command to display the process results.

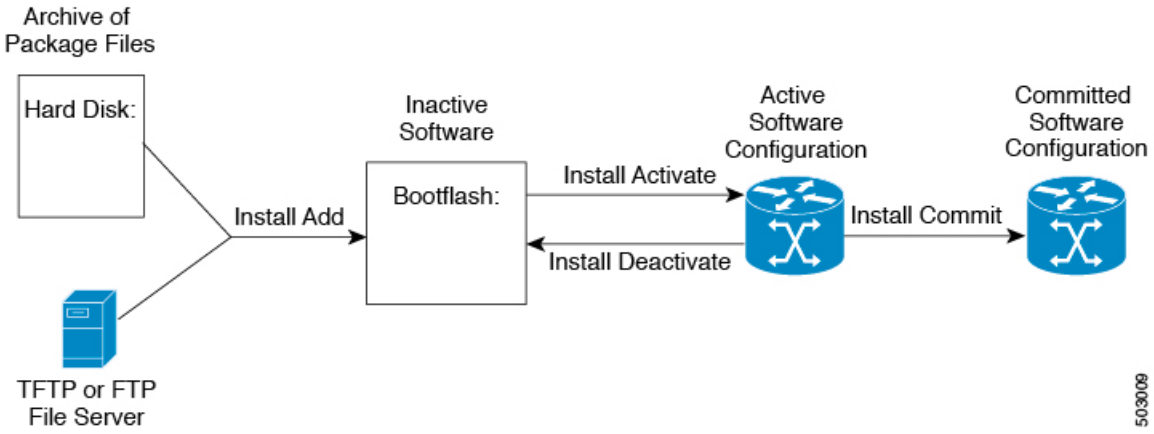
Package Management

The general procedure for adding and activating SMU packages on the device is as follows:

1. Copy the package file or files to a local storage device or file server.
2. Add the package or packages on the device using the **install add** command.
3. Activate the package or packages on the device using the **install activate** command.
4. Commit the current set of packages using the **install commit** command.
5. (Optional) Deactivate and remove the package.

The following figure illustrates the key steps in the package management process.

Figure 13: Process to Add, Activate, and Commit SMU Packages



Guidelines and Limitations

Prerequisites

These prerequisites must be met for a package to be activated or deactivated:

- You must be in a user group associated with a task group that includes the proper task IDs. If you suspect a user group assignment is preventing you from using a command, contact your AAA administrator for assistance.
- Verify that all line cards are installed and operating properly. For example, do not activate or deactivate packages while line cards are booting, while line cards are being upgraded or replaced, or when you anticipate an automatic switchover activity.

Release Specific Guidelines

- Beginning with Cisco NX-OS Release 9.3(9), you can apply a reload SMU along with (ND) ISSU to the same image version (currently running an image on the switch) without a disruptive reload of the switch. You can apply a reload SMU by upgrading to the same image version using ND-ISSU along with the reload SMU using the **install all nxos** <*same image*> **package** <*smu*> **non-disruptive** command.
- Beginning with Cisco NX-OS Release 10.3(4a)M, a new option is added to the **install all nxos** command, **keep-committed-smu** ,to retain the previously installed SMU after upgrade to the same image version when the SMU is compatible with the target image. Execute the command as follows:

install all nxos <*image*> **package** <*smu*> **keep-committed-smu**

Example: `switch# install all nxos nxos64-cs.10.4.3.M.bin package nxos64-cs.CSCxy11111-1.0.0-10.4.3.lib32_64_n9000.rpm keep-committed-smu non-disruptive`

- Beginning with Cisco NX-OS Release 10.5(1)F, the following guidelines are applicable to SMU:
  - If you need to activate an SMU that is valid and compatible with the image, and the activation fails, then the switch reloads automatically. However, if the SMU is not activated after four tries, do not activate the SMU. Meanwhile, when the switch is ready, a syslog message is displayed indicating that the SMU failed to activate.
  - If you try to install a PID-specific SMU on a PID that it is not meant for, then the following message is displayed: **Install operation failed because SMU is not compatible for this switch model** .
  - If you perform ISSU on a switch along with a tar ball of SMUs containing supported and non-supported SMUs, only the supported SMUs will be installed after ISSU.

General Guidelines

- No-reload options are supported in Cisco NX-OS Release 9.3(9) for the SMU installation. The **no-immediate-reload** option is used for activating or deactivating the SMU feature.
- Some packages require the activation or deactivation of other packages. If the SMUs have dependencies on each other, you cannot activate them without first activating the previous ones. SMU packages being activated must be compatible with the image version running in the switch.
- Activation is performed only after the package compatibility checks have been passed. If a conflict is found, an error message is displayed.
- You can install multiple SMU packages by creating a tar bundle. See the Advanced SMU Installation Methods, on page 600 section for more details.
- While a software package is being activated, other requests are not allowed to run on any of the impacted nodes. Package activation is completed when a message similar to this one appears:

  ```
  Install operation 1 completed successfully at Thu Jan 9 01:19:24 2014
  ```

- Each CLI install request is assigned a request ID, which can be used later to review the events.

Prepare for Package Installation

Before you begin

Determine if a software change is required. Verify that the new package is supported on your system. Some software packages require that other packages or package versions be activated, and some packages support only specific line cards.

Review the release notes for important information related to that release and to help determine the package compatibility with your device configuration.

Verify that the system is up, stable, and prepared for the software changes.

You should use several **show** commands to gather information in preparation for the SMU package installation.

Procedure

Step 1 Use this command to verify that the system is ready for SMU package installation, **show logging logfile | grep -i "System ready"**

Example:

```
switch# show logging logfile | grep -i "System ready"
```

Displays if your system is up. Configuring install commands before the system is ready, may result with an "Install operation 11 failed because cannot lock config" error message.

Step 2 Use this command to determine what software should be added on the device and to compare to the active software report after installation operations are complete, **show install active**

Example:

```
switch# show install active
```

Displays the active software on the device.

Step 3 Confirm that all modules are in the stable state using the command **show module**

Example:

```
switch# show module
```

Step 4 Verify that the system clock is correct using the command **show clock**

Example:

```
switch# show clock
```

Software operations use certificates based on device clock times.

This example shows how to verify that the system is up. A "System ready" response indicates that the system is ready for SMU package installation.

```
switch#
    show logging logfile | grep -i "System ready"
2018 Feb 19 11:13:04 switch %ASCII-CFG-2-CONF_CONTROL: System ready
```

This example shows how to display the active packages for the entire system. Use this information to determine if a software change is required.

```
switch#
 show install active
Boot Image:
NXOS Image: bootflash:///nxos.7.0.3.I7.3.1.bin
Active Packages:
switch#
```

This example shows how to display the current system clock setting:

```
switch#
```

```
show clock
02:14:51.474 PST Wed Jan 04 2014
```

Download SMU Package File from Cisco.com

Follow these steps to download the SMU package file:

Procedure

Step 1 Log in to Cisco.com.

Step 2 Go to the Download Software page at this URL: http://software.cisco.com/download/navigator.html

Step 3 In the Select a Product list, choose **Switches** > **Data Center Switches** > **Cisco Nexus 9000 Series Switches** > *model* .

Step 4 Choose the appropriate SMU file for your device and click **Download** .

Copy Package File to Local Storage Device or Network Server

You must copy the SMU package file to a local storage device or a network file server to which the device has access. After this task is done, the package can be added and activated on the device.

If you need to store package files on the device, we recommend that you store the files on the hard disk. The boot device is the local disk from which the package is added and activated. The default boot device is bootflash:.

After the SMU package file has been transferred to a network file server or the local storage device, you are ready to add and activate the file.

Server Protocol Types

Tip Before you copy package files to a local storage device, use the **dir** command to determine if the required package files are already on the device.

If the SMU package files are located on a remote TFTP, FTP, or SFTP server, you can copy the files to a local storage device. After the files are located on the local storage device, the package can be added and activated on the device from that storage device. The following server protocols are supported:

- Trivial File Transfer Protocol—TFTP allows files to be transferred from one computer to another over a network, usually without the use of client authentication (for example, username and password). It is a simplified version of FTP.

Note Some package files might be larger than 32 MB, and the TFTP services provided by some vendors might not support a file this large. If you do not have access to a TFTP server that supports files larger than 32 MB, download the file using FTP.

- File Transfer Protocol—FTP is part of the TCP/IP protocol stack and requires a username and password.
- SSH File Transfer Protocol—SFTP is part of the SSHv2 feature in the security package and provides for secure file transfers. For more information, see the Cisco Nexus 9000 Series NX-OS Security Configuration Guide.

Note Consult your system administrator for the location and availability of your network server.

Use the commands in the following table to copy the SMU package file from the server to your device using the file transfer protocols.

Table 42: Commands for Copying SMU Package Files to the Device

Command	Purpose
copy tftp://*hostname-or-ipaddress*/*directory-path*/*filename* **bootflash:**	Copies the package file from the TFTP server to the bootflash:. • *hostname-or-ipaddress*—The hostname or IP address of the network file server. • *directory-path*—The network file server path that leads to the package file to be added. • *filename*—The name of the package file that you want to add.

Command	Purpose
copy ftp://*username***:***password***@***hostname-or-ipaddress***/***directory-path***/***filename* **bootflash:**	Copies the package file from the FTP server to the bootflash:. • *username*—The username of the user who has access privileges to the directory in which the package file is stored. • *password*—The password associated with the username of the user who has access privileges to the directory in which the package file is stored. If a password is not provided, the networking device accepts anonymous FTP. • *hostname-or-ipaddress*—The hostname or IP address of the network file server. • *directory-path*—The network file server path that leads to the package file to be added. The specified directory should be a directory under the home directory of the user. In this example, the file being downloaded is in a subdirectory called "images" in the home directory of the user "john." **Note** For FTP services, *directory-path* is the directory relative to the *username* home directory. If you want to specify an absolute path for the directory, you must add a "/" following the server address. •

Add and Activate Package

Add and activate SMU package files on your device using the Cisco NX-OS CLI.

You can add SMU package files that are stored on a local storage device or on a remote TFTP, FTP, or SFTP server to your device.

Note The SMU package being activated must be compatible with the currently active software to operate. When an activation is attempted, the system runs an automatic compatibility check to ensure that the package is compatible with the other active software on the device. If a conflict is found, an error message displays. The activation is performed only after all compatibility checks have been passed.

Note Activating an SMU does not cause any earlier SMUs or the package to which the SMU applies to be automatically deactivated.

Before you begin

Make sure that all packages to be added are present on a local storage device or a network file server.

Make sure that you meet all of the prerequisites for the activation of packages.

Complete the procedure described in Copy Package File to Local Storage Device or Network Server, on page 588 .

Procedure

Step 1 Connect to the console port and log in.

Establishes a CLI management session to the console port.

Step 2 (Optional) Display the package files that are available to be added using the command **dir bootflash:**

Step 3 **install add** *filename* [**activate**]

Example:

Unpacks the package software files from the local storage device or network server and adds them to the bootflash: and all active and standby supervisors installed on the device.

Note Only SMU package files can be added and activated using this procedure.

The *filename* argument can take any of these formats:

- **bootflash:** *filename*
- **tftp://** *hostname-or-ipaddress* / *directory-path* / *filename*
- **ftp://** *username* **:** *password* **@** *hostname-or-ipaddress* / *directory-path* / *filename*
- **usb1:** *filename*
- **usb2:** *filename*

For all SMU packages except the CSCur02700 SMU package, you can use the optional **activate** keyword to automatically activate the package after it is added successfully.

Note For the CSCur02700 SMU package, use the **install activate** command in Step 5 to activate the package. Do not use the optional **activate** keyword with the **install add** command as the package might fail and require a reboot.

Multiple versions of an SMU package can be added to the storage device without impacting the running configuration, but only one version of a package can be activated for a line card.

Note Press **?** after a partial package name to display all possible matches available for activation. If there is only one match, press the **Tab** key to fill in the rest of the package name.

Step 4 (Optional) Display the inactive packages on the device using the package using the command **show install inactive**

Example:

```
switch# show install inactive
```

This is an optional step. Verify that the package added in the previous step appears in the display.

Step 5 Required: **install activate** *filename*

Example:

Example:

Activates a package that was added to the device. SMU packages remain inactive until activated. (Skip this step if the package was activated earlier with the **install add activate** command.)

Tip After the activation process finishes, enter the **show install log** command to display the process results.

Step 6 Repeat Step 5 until all packages are activated.

Activates additional packages as required.

Step 7 (Optional) Use this command to determine if the correct packages are active, **show install active**

Example:

```
switch# show install active
```

Displays all active packages. This is an optional step.

Commit Active Package Set

Committing the active package set ensures that the activated SMU packages remain persistent across system reloads.

When an SMU package is activated on the device, it becomes part of the current running configuration. To make the package activation persistent across system-wide reloads, you must commit the package on the device.



Note On startup, the device loads the committed package set. If the system is reloaded before the current active package is committed, the previously committed package set is used.

Before you begin

Before you commit a package set, verify that the device is operating correctly and is forwarding packets as expected.

Complete the procedure described in Add and Activate Package, on page 590 .

Procedure

Step 1 Commit the current set of packages so that these packages are used if the device is restarted using the command **install commit** *filename*

Example:

Step 2 (Optional) Display which packages are committed using the command **show install committed**

Example:

```
switch# show install committed
```

This is an optional step.

This example shows how to commit active SMU packages on the device and then verify the committed packages:

Deactivate and Remove Packages

Use this procedure to deactivate and remove software packages (SMUs) from the device.

When a package is deactivated, it is no longer active on the device, but the package files remain on the boot disk. The package files can be reactivated later, or they can be removed from the disk.

The Cisco NX-OS software also provides the flexibility to roll back the selected package set to a previously saved package set. If you find that you prefer a previous package set over the currently active package set, you can use the **install deactivate** and **install commit** commands to make a previously active package set active again.

Before you begin

You cannot deactivate a package if it is required by another active package. When you attempt to deactivate a package, the system runs an automatic check to ensure that the package is not required by other active packages. The deactivation is performed only after all compatibility checks have been passed.

You cannot delete a package if it is part of the running or committed software of the device.

Procedure

Step 1 Connect to the console port and log in.

Establishes a CLI management session to the console port.

Step 2 Deactivate a package that was added to the device and turns off the package features for the line card using the command **install deactivate** *filename*

Example:

Note You must run **install commit** after **install deactivate** to deactivate the package completely, otherwise the package gets activated again after reload. For reload SMU, run **install commit** after the device reloads.

Step 3 (Optional) Display the inactive packages on the device using the command **show install inactive**

Example:

```
switch# show install inactive
```

This is an optional step.

Step 4 Commit the current set of packages so that these packages are used if the device is restarted using the command **install commit**

Example:

```
switch# install commit
```

This is an optional step.

Note Packages can be removed only if the deactivation operation is committed.

Step 5 **install remove** { *filename* | **inactive** }

Example:

Example:

```
switch# install remove inactive
 Proceed with removing? (y/n)? [n] y
```

Removes the inactive package.

- Only inactive packages can be removed.

- Packages can be removed only if they are deactivated from all line cards in the device.
- The package deactivation must be committed.
- To remove a specific inactive package from a storage device, use the **install remove** command with the *filename* argument.
- To remove all inactive packages from all nodes in the system, use the **install remove** command with the **inactive** keyword.

No-Reload Options for SMU Installation

The following are the no-reload options for SMU installation:

Method 1: CLI Install Add/Activate

```
                switch#


                    show version internal build-identifier

nxos image file: bootflash:///nxos64.10.2.0.184.bin :  S184
switch#
                show install inactive
Boot Image:
        NXOS Image: bootflash:///nxos64.10.2.0.184.bin
Inactive Packages:
Inactive Base Packages:
        tahusd_common-1.0.0.0-10.2.0.184.lib32_64_n9000
        tor-2.0.0.0-10.2.0.184.lib32_n9000
        tor_n9k-2.0.0.0-10.2.0.184.lib32_n9000
switch#
switch#
                install add nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000.rpm
[####################] 100%
Install operation 3 completed successfully at Mon Jul 12 11:32:28 2021
switch#
                show install inactive
Boot Image:
        NXOS Image: bootflash:///nxos64.10.2.0.184.bin
Inactive Packages:
        nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000 available
Inactive Base Packages:
        tahusd_common-1.0.0.0-10.2.0.184.lib32_64_n9000
        tor-2.0.0.0-10.2.0.184.lib32_n9000
        tor_n9k-2.0.0.0-10.2.0.184.lib32_n9000
switch#
switch#
               show install pkg-info nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000
Request timedout:: Success
Name          :  nxos64.CSCaa12345-n9k_ALL
Version       :  1.0.0
Release       :  10.2.1
License       :  Cisco proprietary
Patch Type    :  reload
Requires      :  core
```

```
Provides      :  nxos64.CSCaa12345-n9k_ALL
Conflicts     :
Description   :  This is a patch for CSCaa12345-n9k_ALL
switch#
```

CLI Install Activate PATCH with no-immediate-reload option

```
               switch#
               install activate nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000 ?
  <CR>
  WORD                 Package Name
  forced               Non-interactive
  no-immediate-reload  Skip immediate reload for reload type patches.
switch# install activate nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000
no-immediate-reload
[####################] 100%
Install operation 4 !!WARNING!! This patch will get activated only after
a reload of the switch. at Mon Jul 12 11:33:50 2021
switch#
switch#
               show install inactive
Boot Image:
        NXOS Image: bootflash:///nxos64.10.2.0.184.bin
Inactive Packages:
        nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000 activate_pending_reload
Inactive Base Packages:
        tahusd_common-1.0.0.0-10.2.0.184.lib32_64_n9000
        tor-2.0.0.0-10.2.0.184.lib32_n9000
        tor_n9k-2.0.0.0-10.2.0.184.lib32_n9000
switch#
switch#
               show install patch
Boot Image:
        NXOS Image: bootflash:///nxos64.10.2.0.184.bin
--------------------------------------------------
nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000 Inactive Committed
(activate_pending_reload)
--------------------------------------------------
switch##
switch#
               reload
This command will reboot the system. (y/n)?  [n] y
CISCO SWITCH Ver7.69
Switch G2
Device detected on 0:1:2 after 0 msecs
Device detected on 0:1:1 after 0 msecs
Device detected on 0:1:0 after 0 msecs
…..
```

After switch reload, wait for system in ready state

```
               :///nxos64.10.2.0.184.bin :  S184
switch#
switch#
               show logging logfile | include ready
2021 Jul 12 11:40:34 N93180-1 %ASCII-CFG-2-CONF_CONTROL: System ready
switch#
switch#
               show install patch
Boot Image:
        NXOS Image: bootflash:///nxos64.10.2.0.184.bin
```

```
-----------------------------------------------------
nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000 Active
-----------------------------------------------------
switch#
switch#
                show install active
Boot Image:
        NXOS Image: bootflash:///nxos64.10.2.0.184.bin
Active Packages:
        nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000 active
Active Base Packages:
…..
```

CLI Install Deactivate PATCH with no-immediate-reload option

```
                switch#
                install deactivate nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000 ?
  <CR>
  WORD                 Package Name[Note: startup configuration may get affected]
  forced               Non-interactive
  no-immediate-reload  Skip immediate reload for reload type patches.
switch#
                install deactivate nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000
no-immediate-reload
[####################] 100%
Install operation 5 !!WARNING!! This patch will get deactivated only after
a reload of the switch. at Mon Jul 12 11:42:24 2021
switch#
switch#
                show install patch
Boot Image:
        NXOS Image: bootflash:///nxos64.10.2.0.184.bin
-----------------------------------------------------
nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000 Active (deactivate_pending_reload)
-----------------------------------------------------
switch#
switch#
                show install active
Boot Image:
        NXOS Image: bootflash:///nxos64.10.2.0.184.bin
Active Packages:
        nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000 active
Active Base Packages:
…..
switch#
                reload
WARNING: Uncommitted patches present
This command will reboot the system. (y/n)?  [n] y
CISCO SWITCH Ver7.69
Switch G2
Device detected on 0:1:2 after 0 msecs
Device detected on 0:1:1 after 0 msecs
Device detected on 0:1:0 after 0 msecs
…..
```

After switch reload, wait for system in ready state

```
                switch#
                show logging logfile | include ready
2021 Jul 12 11:52:28 N93180-1 %ASCII-CFG-2-CONF_CONTROL: System ready
switch#
```

```
switch#
                show install patch
Boot Image:
        NXOS Image: bootflash:///nxos64.10.2.0.184.bin
----------------------------------------------------
nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000 Inactive Committed
----------------------------------------------------
switch#
                show install inactive
Boot Image:
        NXOS Image: bootflash:///nxos64.10.2.0.184.bin
Inactive Packages:
        nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000 available
Inactive Base Packages:
        tahusd_common-1.0.0.0-10.2.0.184.lib32_64_n9000
        tor-2.0.0.0-10.2.0.184.lib32_n9000
        tor_n9k-2.0.0.0-10.2.0.184.lib32_n9000
switch#
switch#
                install remove nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000
Proceed with removing nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000? (y/n)?  [n] y
[####################] 100%
Install operation 6 completed successfully at Mon Jul 12 11:57:06 2021
switch#
                show install patch
Boot Image:
        NXOS Image: bootflash:///nxos64.10.2.0.184.bin
----------------------------------------------------
----------------------------------------------------
switch#
                show install inactive
Boot Image:
        NXOS Image: bootflash:///nxos64.10.2.0.184.bin
Inactive Packages:
Inactive Base Packages:
        tahusd_common-1.0.0.0-10.2.0.184.lib32_64_n9000
        tor-2.0.0.0-10.2.0.184.lib32_n9000
        tor_n9k-2.0.0.0-10.2.0.184.lib32_n9000
switch#
```

CLI Install ADD ACTIVATE via bootflash: with no-immediate-reload

```
                switch#
                install add nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000.rpm
activate ?
  <CR>
  downgrade            Downgrade package
  forced               Non-interactive
  no-immediate-reload  Skip immediate reload for reload type patches.
  upgrade              Upgrade package
switch#
                install add nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000.rpm
activate no-immediate-reload
Adding the patch (/nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000.rpm)
[####################] 100%
Install operation 7 completed successfully at Mon Jul 12 12:03:02 2021
Activating the patch (/nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000.rpm)
[####################] 100%
Install operation 8 !!WARNING!! This patch will get activated only after
a reload of the switch. at Mon Jul 12 12:03:10 2021
switch#
```

CLI Install ADD ACTIVATE via tftp with no-immediate-reload

```
               switch#
               install add
tftp://172.27.250.42/auto/tftp-sjc-users1/shuojiun/nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000.rpm
 vrf management activate ?
  <CR>
  downgrade            Downgrade package
  forced               Non-interactive
  no-immediate-reload  Skip immediate reload for reload type patches.
  upgrade              Upgrade package
switch#
               install add
tftp://172.27.250.42/auto/tftp-sjc-user1/tester/nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000.rpm
 vrf management activate no-immediate-reload
[####################] 100%
Install operation 11 !!WARNING!! This patch will get activated only after
a reload of the switch. at Mon Jul 12 12:06:49 2021
switch#
```

Method 2: VIA DME REST Action/Exec payload

Note In the payload below, "reloadFlag": "noreload", you need to set “reloadFlag” as “noreload”. “reloadFlag” is not new in the Action/Exec items.

```
POST URL:
http://172.27.250.239//api/mo/sys/action.json
{
    "actionLCont": {
        "children": [
            {
                "actionLSubj": {
                    "attributes": {
                        "dn": "sys/action/lsubj-[sys]"
                    },
                    "children" : [
                        {
                            "topSystemSwpkgsInstallLTask": {
                                "attributes": {
                                    "dn":
"sys/action/lsubj-[sys]/topSystemSwpkgsInstallLTask",
                                    "pkgAction": "add-activate",
                                    "reloadFlag": "noreload",
                                    "adminSt": "start",
                                    "url":
"nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000.rpm"
                                }
                            }
                        }
                    ]
                }
            }
        ]
    }
}
{
    "actionLCont": {
        "children": [
            {
```

```
                "actionLSubj": {
                    "attributes": {
                        "dn": "sys/action/lsubj-[sys]"
                    },
                    "children" : [
                        {
                            "topSystemSwpkgsInstallLTask": {
                                "attributes": {
                                    "dn":
"sys/action/lsubj-[sys]/topSystemSwpkgsInstallLTask",
                                    "pkgAction": "activate",
                                    "reloadFlag": "noreload",
                                    "adminSt": "start",
                                    "url":
"nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000"
                                }
                            }
                        }
                    ]
                }
            }
        ]
    }
}
{
    "actionLCont": {
        "children": [
            {
                "actionLSubj": {
                    "attributes": {
                        "dn": "sys/action/lsubj-[sys]"
                    },
                    "children" : [
                        {
                            "topSystemSwpkgsInstallLTask": {
                                "attributes": {
                                    "dn":
"sys/action/lsubj-[sys]/topSystemSwpkgsInstallLTask",
                                    "pkgAction": "deactivate",
                                    "reloadFlag": "noreload",
                                    "adminSt": "start",
                                    "url":
"nxos64.CSCaa12345-n9k_ALL-1.0.0-10.2.1.lib32_64_n9000"
                                }
                            }
                        }
                    ]
                }
            }
        ]
    }
}
```

Advanced SMU Installation Methods

Install Multiple SMU Packages Using a Single TAR File

If you want to install multiple SMU packages, you can create a single TAR bundle file and use it across the switches in the Data Center.

Follow these steps to generate a TAR file from a given list of SMU packages downloaded from a software download center.

Note The file name mentioned in the following examples is for illustration purposes only, and the actual file name will depend on the appropriate release.

- Step 1 - Create a new folder in the user computer or virtual machine.

```
bash#
mkdir nx1043
```

- Step 2 - Download the required SMU packages from the Cisco Software Download Center portal and copy the SMU packages to the new folder.

```
bash#
cp nxos64-cs.CSCxy11111-1.0.0-10.4.3.lib32_64_n9000.rpm  nx1043/

bash#
cp nxos64-cs.CSCxy22222-1.0.0-10.4.3.lib32_64_n9000.rpm  nx1043/
```

- Step 3 - Create a tar bundle file.

```
bash#
cd nx1043
bash#
tar cf nxos64-cs.10.4.3.smu.bundle.tar *.rpm
```

- Step 4 - Use the existing **install add** *filename* **activate** command to install the SMU packages from the TAR bundle.

```
switch#
install add nxos64-cs.10.4.3.smu.bundle.tar activate
```

Install SMU Packages as Part of the New NX-OS Software Image Installation

On Cisco Nexus switches, the NX-OS software image can be upgraded to a newer version using the **install all** command. This command has been enhanced to include SMU packages separate from the NX-OS switch software image, which benefits the software maintenance by reducing the number of reload required during the installation process for both the software image and SMU packages.

On Cisco Nexus switches, the NX-OS software image can be upgraded to a newer version using the install all command. This command has been enhanced to include SMU packages seperate from the NX-OS switch software image, which benefits software maintenance by reducing the number of reloads required during the installation process for both the software image and SMU packages.

The **install all** command can be initiated with a single .tar bundle file containing either:

- One NX-OS software image and a single SMU.rpm file

- One NX-OS software image and a tar bundle of multiple SMU.rpm files

When the **install all** command is initiated with one or more SMU .rpm files, the switch will automatically commit the SMU files after the upgrade.

If the switch is reloaded during bootup, the SMUs will not be applied and will remain in an inactive state. The SMUs can be installed using the **install all** or **install activate** commands.

The following section describes all the supported scenarios when SMU packages are included in the **install all** command.

Note The child tar bundle must not contain a mix of SMU .rpm files and another tar bundle of SMU .rpm files.

Note The filename mentioned in the following examples is for illustration purposes only, and the actual filename will depend on the appropriate release.

Example-1 : In this scenario, a new software image and a single SMU package is used.

```
switch#
install all nxos nxos64-cs.10.4.3.M.bin package
nxos64-cs.CSCxy11111-1.0.0-10.4.3.lib32_64_n9000.rpm
```

Example-2 : In this scenario, a set of SMU packages are created as a TAR bundle following the TAR file method mentioned above and installed along with the NX-OS software image.

```
switch#
install all nxos nxos64-cs.10.4.3.M.bin package
nxos64-cs.10.4.3.smu.bundle.tar
```

Example-3 : In this scenario, a single SMU package and the NX-OS software image can be bundled into one single tar file and installed using the **install all** command.

```
switch#
install all nxos nxos64-cs.10.4.3.M.SMU.plus.IMAGE.tar
```

1. Download the SMU package from the Cisco Download center. For example: nxos64-cs.CSCxy11111-1.0.0-10.4.3.lib32_64_n9000.rpm
2. Download **nxos64-cs.10.4.3.M.bin** and place it in the same folder.
3. Create a tar bundle **nxos64-cs.10.4.3.M.SMU.plus.IMAGE.tar** consisting of the NX-OS image and SMU package.

```
bash#
tar cf nxos64-cs.10.4.3.M.SMU.plus.IMAGE.tar
nxos64-cs.10.4.3.M.bin nxos64-cs.CSCxy11111-1.0.0-10.4.3.lib32_64_n9000.rpm
```

Example-4 : When there are multiple SMU packages must be installed along with the NX-OS image, the SMU packages must be built into a SMU tar bundle file first as explained in the Install Multiple SMU Packages Using a Single TAR File section. Subsequently, this SMU tar bundle can be further bundled together with the NX-OS image and a single tar file could be used in the **install all** command.

```
Switch#
install all nxos nxos64-cs.10.4.3.M.SMU.BUNDLE.plus.IMAGE.tar
```

1. Create a SMU tar bundle image with the list of SMU packages as explained in the Install Multiple SMU Packages Using a Single TAR File section.

```
                                bash#
                                mkdir nx1043
bash#
                                cp nxos64-cs.CSCxy11111-1.0.0-10.4.3.lib32_64_n9000.rpm
nx1043/
bash#
                                cp nxos64-cs.CSCxy22222-1.0.0-10.4.3.lib32_64_n9000.rpm
nx1043/
bash#
                                cd nx1043
bash#
                                tar cf nxos64-cs.10.4.3.smu.bundle.tar *.rpm
```

2. Download **nxos64-cs.10.4.3.M.bin** and place it in the same folder.
3. Create another tar bundle **nxos64-cs.10.4.3.M.SMU.BUNDLE.plus.IMAGE.tar** .

```
                                bash#
                                tar cf nxos64-cs.10.4.3.M.SMU.BUNDLE.plus.IMAGE.tar
nxos64-cs.10.4.3.M.bin nxos64-cs.10.4.3.smu.bundle.tar
```

Downgrade Feature RPMs

Follow this procedure to downgrade an installed feature RPM to the base feature RPM.

Procedure

Step 1 Display the feature RPM packages on the device using the command **show install packages**

Example:

```
switch# show install packages
                      ntp.lib32_n9000     1.0.1-7.0.3.I2.2e      installed
```

This is an optional step.

Step 2 Load Bash using the command **run bash**

Example:

```
switch# run bash
                        bash-4.2$
```

Step 3 Change to the RPMs folder in Bash using the command **cd /rpms**

Example:

```
bash-4.2$ cd /rpms
```

Step 4 List the RPM for the specified feature using the command **ls *** *feature* *****

Example:

```
bash-4.2$ ls *ntp*
                        ntp-1.0.0-7.0.3.I2.2e.lib32_n9000.rpm
```

Step 5 Copy the base feature RPM to the bootflash using the command **cp** *filename* **/bootflash**

Example:

```
bash-4.2$ cp ntp-1.0.0-7.0.3.I2.2e.lib32_n9000.rpm /bootflash
```

Step 6 Exit Bash using the command **exit**

Example:

```
bash-4.2$ exit
```

Step 7 Required: Downgrade the feature RPM using the command **install add bootflash:** *filename* **activate downgrade**

Example:

```
switch# install add bootflash:ntp-1.0.0-7.0.3.I2.2e.lib32_n9000.rpm activate downgrade
                        Adding the patch (/ntp-1.0.0-7.0.3.I2.2e.lib32_n9000.rpm)
                        [#############       ]  60%
                        Adding the patch (/ntp-1.0.0-7.0.3.I2.2e.lib32_n9000.rpm)
                        [####################] 100%
                        Install operation 11 completed successfully at Thu Sep  8 15:35:35 2015
                        Activating the patch (/ntp-1.0.0-7.0.3.I2.2e.lib32_n9000.rpm)
                       This install operation requires system reload. Do you wish to continue (y/n)?:
 [n] y
                        [  217.975959] [1473348971]  writing reset reason 132, System reset due to
reload patch(es) activation
                        [  217.991166] [1473348971]□□
                        CISCO SWITCH Ver7.51
                        Device detected on 0:6:0 after 0 msecs
                        Device detected on 0:1:1 after 0 msecs
                        Device detected on 0:1:0 after 0 msecs
                        MCFrequency 1333Mhz
                        Relocated to memory
```

Note If you are prompted to reload the device, enter **Y** . A reload is required only when downgrading the NTP and SNMP feature RPMs.

Step 8 Display the base feature RPM on the device using the command **show install packages | i** *feature*

Example:

```
switch# show install packages | i ntp
                        ntp.lib32_n9000    1.0.0-7.0.3.I2.2e   installed
```

This is an optional step.

Display Installation Log Information

The installation log provides information on the history of the installation operations. Each time an installation operation is run, a number is assigned to that operation.

- Use the **show install log** command to display information about both successful and failed installation operations.
- Use the **show install log** command with no arguments to display a summary of all installation operations. Specify the *request-id* argument to display information specific to an operation. Use the **detail** keyword to display details for a specific operation, including file changes, nodes that could not be reloaded, and any impact to processes.

Configuration Examples

This example shows how to display information for all installation requests:

This example shows how to display additional information, including any impact to nodes and processes:

This example shows the output after an SMU package has been activated but before the switch has been reloaded:

Perform Software Maintenance Upgrade for Guest Shell Bash

You can perform a software maintenance upgrade for Bash in the Guest Shell.

Procedure

Step 1 Download the SMU package file for Guest Shell Bash from Cisco.com.

Obtains the package file from Cisco.com. For instructions, see Download SMU Package File from Cisco.com, on page 588 .

Step 2 Copy the SMU package file to the bootflash: of the switch.

Copies the package file to the device. For instructions, see Copy Package File to Local Storage Device or Network Server, on page 588 .

Step 3 Access the Guest Shell using the command **guestshell**

Example:

```
switch# guestshell
 guestshell:~$
```

Step 4 Upgrade the existing Bash file in the Guest Shell using the command **sudo rpm -Uvh** / bootflash / *filename*

Example:

```
guestshell:~$ sudo rpm -Uvh /bootflash/bash-4.2-r8.x86_64.rpm
 Preparing...                ########################################### [100%]
 1:bash                   ########################################### [100%]
 update-alternatives: Linking //bin/sh to /bin/bash
```

Step 5 Verify that the new version of the Bash file was installed successfully using the command **rpm -qa | grep bash**

Example:

```
guestshell:~$ rpm -qa | grep bash
 bash-4.2-r8.x86_64
```

Step 6 **guestshell sync**

Example:

```
switch# guestshell sync
 Access to the guest shell will be temporarily disabled while it synchronizes contents to standby.
 Are you sure you want to continue? (y/n) [n] y
 dt-n9k3-1# 2014 Oct  7 05:00:01 dt-n9k3-1 %$ VDC-1 %$ %VMAN-2-INSTALL_STATE: Deactivating virtual
service 'guestshell+'
 dt-n9k3-1# 2014 Oct  7 05:00:06 dt-n9k3-1 %$ VDC-1 %$ %VMAN-2-ACTIVATION_STATE: Successfully
deactivated virtual
 service 'guestshell+'
 2014 Oct  7 05:00:12 dt-n9k3-1 %$ VDC-1 %$ %VMAN-2-ACTIVATION_STATE: Successfully deactivated virtual
 service
 'guestshell+' ; Starting sync to standby sup
 2014 Oct  7 05:00:32 dt-n9k3-1 %$ VDC-1 %$ %VMAN-2-MOVE_STATE: Successfully synced virtual service
 'guestshell+' ;
 Activating
 2014 Oct  7 05:00:32 dt-n9k3-1 %$ VDC-1 %$ %VMAN-2-ACTIVATION_STATE: Activating virtual service
'guestshell+'
 2014 Oct  7 05:00:56 dt-n9k3-1 %$ VDC-1 %$ %VMAN-2-ACTIVATION_STATE: Successfully activated virtual
 service
 'guestshell+'
```

On a dual-supervisor system, synchronizes the rootfs with the Bash SMU version to the standby supervisor before doing a switchover. If you do not run this command, you will need to repeat this procedure after a supervisor switchover.

Note The new Bash file is preserved after a Guest Shell reboot or Guest Shell disable+enable. However, you need to reinstall the Guest Shell Bash SMU package file after a Guest Shell destroy+enable.

Related Documents

This reference provides information about related documents for software upgrades.

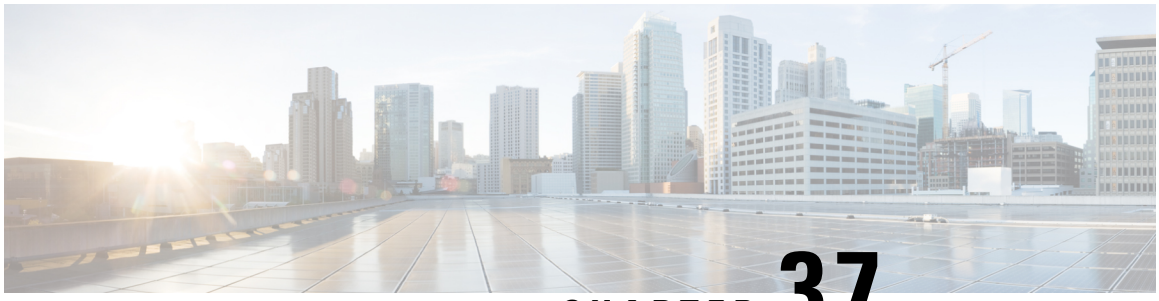
Related Topic	Document Title
Software upgrades	*Cisco Nexus 9000 Series NX-OS Software Upgrade and Downgrade Guide*

History

This table lists the release history for SMU package files.

Table 43: SMU Package File Release History

SMU Package File	Releases	Description
bash-4.2-r8.x86_64.rpm	6.1(2)I3(1)	Guest Shell Bash SMU for Bash vulnerabilities CVE-2014-6277, CVE-2014-6278, CVE-2014-7186, and CVE-2014-7187
n9000-dk9.6.1.2.I3.1.CSCur02700.bin	6.1(2)I3(1) and all 6.1(2)I2(x) releases	Cisco NX-OS SMU for CSCur02700 (Bash vulnerabilities CVE-2014-6277, CVE-2014-6278, CVE-2014-7186, and CVE-2014-7187)
n9000-dk9.6.1.2.I2.1.CSCup81353.bin	6.1(2)I2(1), 6.1(2)I2(2), 6.1(2)I2(2a), and 6.1(2)I2(3)	Cisco NX-OS SMU for CSCup81353



CHAPTER 37

Configuration Replace and Commit-timeout

- Overview, on page 609
- Guidelines and Limitations, on page 618
- Perform Configuration Replace, on page 622

Overview

Configuration Replace

The configuration replace feature enables you to replace the running configuration of the Cisco Nexus switch with the user provided configuration without reloading the device. The device reload may be required only when a configuration itself requires a reload. The running configuration file that is provided by the user should be taken using copy running file. Unlike **copy file: to running** , the configuration replace feature is not a merge operation. This feature replaces the entire running configuration with a new configuration that is provided by the user. If there is a failure in the configuration replace, the original configuration is restored in the switch. From Cisco NX-OS Release 9.3(1), **best-effort** option is introduced. This option enables the configuration replace to execute the full patch despite any error in the commands and the original configuration is not restored in the switch.

Commit-Timeout

The commit-timeout feature enables you to rollback to the previous configuration after successfully performing the configuration replace operation. If the commit timer expires, the rollback operation is automatically initiated.

Operation Steps

Note

- You must provide a valid running configuration that has been received with the Cisco NX-OS device. It should not be a partial configuration.

The configuration replace feature has the following operation steps:

- Configuration replace intelligently calculates the difference between the current running-configuration and the user-provided configuration in the Cisco Nexus switch and generates a patch file which is the

difference between the two files. You can view this patch file which includes a set of configuration commands.

- Configuration replace applies the configuration commands from the patch file similarly to executing commands.
- The configuration rolls back to or restores the previous running configuration under the following situations:
  - If there is a mismatch in the configuration after the patch file has been applied.
  - If you perform the configuration operation with a commit timeout and the commit timer expires.
- The configuration does not roll back to or does not restore the previous running configuration when the best-effort option is used. This option enables the configuration replace to execute the full patch despite any error in the commands and will not roll back to the previous configuration.
- You can view the exact configuration that caused a failure using the **show config-replace log exec** command.
- Restore operations that fail while restoring the switch to the original configuration, are not interrupted. The restore operation continues with the remaining configuration. Use the **show config-replace log exec** command to list the commands that failed during the restore operation.
- If you enter the **configure replace commit** command before the timer expires, the commit timer stops and the switch runs on the user provided configuration that has been applied through the configuration replace feature.
- If the commit timer expires, roll back to the previous configuration is initiated automatically.
- In Cisco NX-OS Release 9.3(1), semantic validation support is added for the configuration replace. This semantic validation is done as part of the precheck in configuration replace. The patch gets applied only when the semantic validation is successful. After applying the patch file, configuration replace triggers the verification process. The configuration replace compares the running-configuration with the user configuration file during the verification process. If there is a mismatch, it restores the device to the original configuration.

Table 44: DIfference between Configuration replace and copying a file to the running-configuration are as follows

Configuration Replace	Copying a file
The configure replace <*target-url*> command removes the commands from the current running-configuration that are not present in the replacement file. It also adds commands that need to be added to the current running-configuration.	The **copy** <*source-url*> **running-config** command is a merge operation which preserves all the commands from, both the source file and the current running-configuration. This command does not remove the commands from the current running-configuration that are not present in the source file.
You must use a complete Cisco NX-OS configuration file as the replacement file for the configure replace <*target-url*> command.	You can use a partial configuration file as a source file for the **copy** <*source-url*> **running-config** command.

Benefits

- You can replace the current running-configuration file with the user-provided configuration file without having to reload the switch or manually undo CLI changes to the running-configuration file. As a result, the system downtime is reduced.
- You can revert to the saved Cisco NX-OS configuration state.
- It simplifies the configuration changes by allowing you to apply a complete configuration file to the device, where only the commands that need to be added or removed are affected. The other service and configurations that are not modified remain untouched.
- If you configure the commit-timeout feature, you can rollback to the previous configuration even when the configuration replace operation has been successful.

Recommended Workflow



Note
- This workflow needs to be the same in the candidate config.
- Default configuration in the candidate config is not supported.

1. Generate a configuration file by first applying the configurations on a Cisco Nexus Series device and then use the **show running-configuration** output as the configuration file. Use this file to make configuration modifications as required. Then use this generated or updated configuration file to perform configuration replace.
2. View and verify the patch file by executing the **configure replace** <*file*> **show-patch** command. This is an optional step.
3. Run the configuration replace file either using or skipping the **commit-timeout** <*time*> feature. Based on your requirements, you can perform one of the following steps:
   - Run **configure replace** <*file*> **verbose** to see the commands that get executed with configuration replace on the console.
   - Run the **configure replace** [**bootflash**/**scp**/**sftp**] <*user-configuration-file*> **verbose commit-timeout** <*time*> commands to configure the commit time.
4. Run the **configure replace commit** command to stop the commit timer. This step is necessary if you have run the configuration replace operation with the commit-timeout feature.
5. Configuration replace performs a precheck that includes the semantic validation of the configuration. The configuration replace operation fails if there is an error. Use the **show config-replace log verify** command to see the details of the failed configurations. After applying the patch file, configuration replace triggers the verification process. The configuration replace compares the running-configuration with the user configuration file during the verification process. If there is a mismatch, it restores the device to the original configuration. Use the **show config-replace log verify** command to see the mismatched configurations.
6. You can perform the following configuration replace operations in Cisco NX-OS Release 9.3(1):
   - Configuration replace without the semantic validation and without best-effort mode.
   - Configuration replace without the semantic validation and with best-effort mode.

- Configuration replace with the semantic validation and without best-effort mode.
- Configuration replace with the semantic validation and with best-effort mode.

Verification

To check and verify configuration replace and its status, use the commands that are outlined in the table:

Table 45: Verifying Configuration Replace

Command	Purpose
configure replace [**bootflash**/**scp**/**sftp**]<*user-configuration-file*] **show-patch**	Displays the difference between the running-configurations and user-provided configurations.
show config-replace log exec	Displays a log of all the configurations executed and those that failed. In case of an error, it displays an error message against that configuration.
show config-replace log verify	Displays the configurations that failed, along with an error message. It does not display configurations that were successful.
show config-replace status	Displays the status of the configuration replace operations, including in-progress, successful, and failure. If you have configured the commit-timeout feature, the commit and timer status and the commit timeout time remaining is also displayed.

Configuration Examples

See the following configuration examples for configuration replace:

- Use the **configure replace bootflash:** <*file*> **show-patch** CLI command to display the difference between the running-configurations and user-provided configurations.

```
switch(config)# configure replace bootflash:<file> show-patch
Collecting Running-Config
Converting to checkpoint file
#Generating Rollback Patch
!!
no role name abc
```

- Use the **configure replace bootflash:** <*file*> **verbose** CLI command to replace the entire running-configuration in the switch with the user-configuration.

```
switch(config)# configure replace bootflash:<file> verbose
Collecting Running-Config
Generating Rollback patch for switch profile
Rollback Patch is Empty
Note: Applying config parallelly may fail Rollback verification
Collecting Running-Config
#Generating Rollback Patch
Executing Rollback Patch
==========================================================
config t
```

```
no role name abc
==========================================================
Generating Running-config for verification
Generating Patch for verification

Rollback completed successfully.

Sample Example with adding of BGP configurations.
 switch(config)# sh run | section bgp
 switch(config)# sh file bootflash:file | section bgp
feature bgp
router bgp 1
    address-family ipv4 unicast
    neighbor 1.1.1.1
switch(config)#
switch(config)# configure replace bootflash:file verbose
Collecting Running-Config
Generating Rollback patch for switch profile
Rollback Patch is Empty
Note: Applying config parallelly may fail Rollback verification
Collecting Running-Config
#Generating Rollback Patch
Executing Rollback Patch
==========================================================
config t
feature bgp
router bgp 1
address-family ipv4 unicast
neighbor 1.1.1.1
==========================================================
Generating Running-config for verification
Generating Patch for verification

Rollback completed successfully.

switch(config)# sh run | section bgp
feature bgp
router bgp 1
  address-family ipv4 unicast
  neighbor 1.1.1.1

Sample Example with ACL
 switch(config)# configure replace bootflash:run_1.txt
 Collecting Running-Config
 Generating Rollback patch for switch profile
 Rollback Patch is Empty
 Note: Applying config parallelly may fail Rollback verification
 Collecting Running-Config
 #Generating Rollback Patch
 Executing Rollback Patch
 ==========================================================
 config t
 no ip access-list nexus-50-new-xyz
 ip access-list nexus-50-new-xyz-jkl-abc
 10 remark Newark
 20 permit ip 17.31.5.0/28 any
 30 permit ip 17.34.146.193/32 any
 40 permit ip 17.128.199.0/27 any
 50 permit ip 17.150.128.0/22 any
 ==========================================================
 Generating Running-config for verification
 Generating Patch for verification

 Rollback completed successfully.
```

```
switch(config)#


switch(config)# show run aclmgr | sec nexus-50-new-xyz-jkl-abc
ip access-list nexus-50-new-xyz-jkl-abc
  10 remark Newark
  20 permit ip 17.31.5.0/28 any
  30 permit ip 17.34.146.193/32 any
  40 permit ip 17.128.199.0/27 any
  50 permit ip 17.150.128.0/22 any
```

- Use the **configure replace bootflash:user-config.cfg verify-only** CLI command to generate and verify the patch semantically.

```
switch(config)# configure replace bootflash:user-config.cfg verify-only

Version match between user file and running configuration.
Pre-check for User config PASSED
Collecting Running-Config
Converting to checkpoint file
Generating Rollback Patch
Validating Patch
==================================================
`config t `
`interface Ethernet1/1`
`shutdown`
`no switchport trunk allowed vlan`
`no switchport mode`
`no switchport`
`exit`
Skip non dme command for CR validation
`interface Vlan1`
`shutdown`
`interface Ethernet1/1`
`shutdown`
`no switchport`
`ip address 1.1.1.1/24`
`exit`
Skip non dme command for CR validation
==========================================
Patch validation completed successful
switch(config)#
```

- Use the **configure replace bootflash:user-config.cfg best-effort verify-and-commit** CLI command to replace the switch running configuration with the given user configuration after performing the sematic validation on patch.

```
switch(config)# configure replace bootflash:user-config.cfg best-effort verify-and-commit


 Version match between user file and running configuration.
Pre-check for User config PASSED
ADVISORY: Config Replace operation started...
Modifying running configuration from another VSH terminal in parallel
is not recommended, as this may lead to Config Replace failure.

Collecting Running-Config
Generating Rollback patch for switch profile
Rollback Patch is Empty
Collecting Running-Config
Generating Rollback Patch

Validating Patch
```

```
Patch validation completed successful
Executing Rollback Patch
During CR operation,will retain L3 configuration
when vrf member change on interface
Generating Running-config for verification
Generating Rollback Patch

Configure replace completed successfully. Please run 'show config-replace log exec' to
 see if there is any configuration that requires reload to take effect.

switch(config)#
```

- Use the **show config-replace log exec** CLI command to check all the configuration that is executed and failures if any.

```
switch(config)# show config-replace log exec
Operation            : Rollback to Checkpoint File
Checkpoint file name : .replace_tmp_28081
Scheme               : tmp
Rollback done By     : admin
Rollback mode        : atomic
Verbose              : enabled
Start Time           : Wed, 06:39:34 25 Jan 2017
--------------------------------------------------------------------------------

time: Wed, 06:39:47 25 Jan 2017
Status: SUCCESS
End Time             : Wed, 06:39:47 25 Jan 2017
Rollback Status      : Success

Executing Patch:
----------------
switch#config t
switch#no role name abc
```

- Use the **show config-replace log verify** CLI command to check the failed configuration if any.

```
switch(config)# show config-replace log verify
Operation            : Rollback to Checkpoint File
Checkpoint file name : .replace_tmp_28081
Scheme               : tmp
Rollback done By     : admin
Rollback mode        : atomic
Verbose              : enabled
Start Time           : Wed, 06:39:34 25 Jan 2017
End Time             : Wed, 06:39:47 25 Jan 2017
Status               : Success

Verification patch contains the following commands:
--------------------------------------------------
!!
! No changes
--------------------------------------------------------------------------------

time: Wed, 06:39:47 25 Jan 2017
Status: SUCCESS
```

- Use the **show config-replace status** CLI command to check the status of configuration replace.

```
switch(config)# show config-replace status
Last operation : Rollback to file
Details:
  Rollback type: atomic replace_tmp_28081
  Start Time: Wed Jan 25 06:39:28 2017
  End Time: Wed Jan 25 06:39:47 2017
```

```
    Operation Status: Success
switch(config)#
```

Configure Replace might fail when the manually created configuration is used instead of the configuration generated from the switch. The reason for possible failures is the potential difference in the default configuration that isn't shown in the show running configuration. Refer to the following examples:

If the power redundant command is the default command, it doesn't get displayed in the default configuration. But it's displayed when you use the **show run all** command. See the following example:

```
switch# show run all

!Command: show running-config all
!Running configuration last done at: Tue Nov 12 11:07:44 2019
!Time: Tue Nov 12 11:16:09 2019

version 9.3(1) Bios:version 05.39
power redundancy-mode ps-redundant
no hardware module boot-order reverse
no license grace-period
<snip>
hostname n9k13
```

The power redundant command isn't shown in the show running configuration command out. See the following example:

```
!Command: show running-config
!Running configuration last done at: Tue Nov 12 11:07:44 2019
!Time: Tue Nov 12 11:17:24 2019

version 9.3(1) Bios:version 05.39
hostname n9k13
```

When the **power redundancy-mode ps-redundant** command is added in the user configuration for the configure replace; then the verification/commit might fail. See the following example:

```
switch# show file bootflash:test

!Command: show running-config
!Running configuration last done at: Tue Nov 12 10:56:49 2019
!Time: Tue Nov 12 11:04:57 2019

version 9.3(1) Bios:version 05.39
power redundancy-mode ps-redundant
hostname n9k13
```

The **power redundancy-mode ps-redundant** command will not be shown in the show running after configure replace; therefore it will be considered as “missing” and the CR will fail. An example is given below.

```
switch# config replace bootflash:test verify-and-commit

Version match between user file and running configuration.
Pre-check for User config PASSED
ADVISORY: Config Replace operation started...
Modifying running configuration from another VSH terminal in parallel
is not recommended, as this may lead to Config Replace failure.

Collecting Running-Config
Generating Rollback patch for switch profile
Rollback Patch is Empty
Collecting Running-Config
.Generating Rollback Patch

Validating Patch
```

```
Patch validation completed successful
Executing Rollback Patch
During CR operation,will retain L3 configuration
when vrf member change on interface
Generating Running-config for verification
Generating Rollback Patch
Executing Rollback Patch
During CR operation,will retain L3 configuration
when vrf member change on interface
Generating Running-config for verification
Generating Patch for verification
Verification failed, Rolling back to previous configuration
Collecting Running-Config
Cleaning up switch-profile buffer
Generating Rollback patch for switch profile
Executing Rollback patch for switch profiles. WARNING - This will change the
configuration of switch profiles and will also affect any peers if configured
Collecting Running-Config
Generating Rollback Patch
Rollback Patch is Empty
Rolling back to previous configuration is successful

Configure replace failed. Use 'show config-replace log verify' or 'show config-replace
log exec' to see reasons for failure

n9k13# show config-replace log verify
Operation : Config-replace to user config
Checkpoint file name : .replace_tmp_31849
Scheme : tmp
Cfg-replace done By : agargula
Cfg-replace mode : atomic
Verbose : disabled
Start Time : Tue, 11:20:59 12 Nov 2019
Start Time UTC : Tue, 10:20:59 12 Nov 2019
------------------------------------------
End Time : Tue, 11:21:28 12 Nov 2019
End Time UTC : Tue, 10:21:28 12 Nov 2019
Status : Failed

Verification patch contains the following commands:
-------------------------------------------------
!!
Configuration To Be Added Missing in Running-config
=================================================
!
power redundancy-mode ps-redundant

Undo Log
--------------------------------------------------------------------------------
End Time : Tue, 11:21:32 12 Nov 2019
End Time UTC : Tue, 10:21:32 12 Nov 2019
Status : Success
n9k13#
```

In the above example, CR will consider the default commands that are missing and will therefore fail.

Guidelines and Limitations

General Guidelines

- The configuration replace feature could fail if the running configuration includes the **feature-set mpls** or the **mpls static range** commands and tries to move to a configuration without MPLS or modifies the label range.
- Only one user can perform the configuration replace, checkpoint, and rollback operations, or copy the running-configuration to the startup configuration at the same time. Parallel operations such as operations via multiple Telnet, SSH, or NX-API sessions are not supported. The multiple configuration replace or rollback request is serialized, for example, only after the first request is completed, processing of the second request begins.
- For a successful configuration replace operation, sequence number must be present for all ACE entries in ACL in the target configuration file.
- The commit-timeout feature is initiated only if you perform the configuration replace operation with the commit-timeout. The timer value range is from 30 to 3600 seconds.
- If the line card to which the configuration replace feature is applied is offline, the configuration replace operation fails.
- An ITD service must be shut down (**shutdown**) prior to making ITD changes with the configuration replace feature.
- Using the **configure replace** command from maintenance mode asks for a user-confirmation with the following warning:

```
Warning: System is in maintenance mode. Please ensure user config won't inadvertently
revert back config in maintenance mode profile.
            Do you wish to proceed anyway? (y/n)  [n]
```

- Sequence number is mandatory for CLI **ip community-list** and **ip as-path access-list** commands. Without a sequence number, the configuration replace operation fails.
- If your configurations demand reloading the Cisco NX-OS device in order to apply the configuration, then you must reload these configurations after the configuration replace operation.

Configuration Replace for PBR Commands

The content of this section is applicable from Cisco NX-OS Release 10.4(3)F.

None of the PBR commands can coexist under the same parent route-map. If the mutually exclusive PBR commands are given under the same route-map in the candidate config, the config-replace patch is generated only for the last command variant under the route-map and is applied after CR operation.

The following table depicts a few use cases.

Use Case	Candidate Config	Converted Candidate Config
Use Case 1: Multiple command variants - Only the last command variant is retained The candidate config is automatically converted as shown in the third column before the CR patch is generated.	route-map rmap1 permit 10 set ip next-hop 1.1.1.1 2.2.2.2 set ipv6 next-hop 3::3 set ip next-hop verify-availability 4.4.4.4 set ip next-hop verify-availability 5.5.5.5 set ip vrf green next-hop 6.6.6.6 set ip vrf blue next-hop 7.7.7.7 8.8.8.8	route-map rmap1 permit 10 set ip vrf green next-hop 6.6.6.6 set ip vrf blue next-hop 7.7.7.7 8.8.8.8
Use Case 2: Commands comprising track IDs - Only the last command variant with same next-hop and different track ID is retained For the verify-availability commands, track ID cannot be modified for the same next-hop. The candidate config is automatically converted as shown in the third column before the CR patch is generated.	route-map test permit 10 set ip next-hop verify-availability 1.1.1.1 track 1 set ip next-hop verify-availability 2.2.2.2 track 20 set ip next-hop verify-availability 2.2.2.2 track 30 set ip next-hop verify-availability 2.2.2.2 track 40 set ip next-hop verify-availability 3.3.3.3 track 3	route-map test permit 10 set ip next-hop verify-availability 1.1.1.1 track 1 set ip next-hop verify-availability 2.2.2.2 track 40 set ip next-hop verify-availability 3.3.3.3 track 3

Commands with Mandatory Sequence Number

Ensure that you provide the sequence number mandatorily for the following commands in the candidate-config file. Without a sequence number the configuration replace operation fails:

- **ip prefix-list list-name seq** *seq* { **deny** | **permit** } *prefix*
- **ipv6 prefix-list list-name seq** *seq* { **deny** | **permit** } *prefix*
- **mac-list list-name seq** *seq* { **deny** | **permit** } *prefix*
- **ip community-list** { **standard** | **expanded** } *list-name* seq *seq* { **deny** | **permit** } *expression*
- **ip extcommunity-list** { **standard** | **expanded** } *list-name* seq *seq* { **deny** | **permit** } *expression*
- **ip large-community-list** { **standard** | **expanded** } *list-name* seq *seq* { **deny** | **permit** } *expression*
- **ip-as-path access-list** *list-name* seq *seq* { **deny** | **permit** } *expression*

Not Supported Features

1. You are not allowed to initiate another configuration replace operation when the commit timer is running. You must either stop the timer by using the **configure replace commit** command or wait until the commit timer expires before you initiate another configuration replace operation.
2. The configuration replace operation is not supported if you attempt to replace a multichassis EtherChannel trunk (MCT) configuration with a virtual peer-link configuration. This operation is not allowed because

the physical MCT uses the CFS distribution over Ethernet mode and the virtual peer-link use the CFS distribution over IP mode.

3. Entering maintenance mode from the user configuration is not supported.
4. The semantic validation is not supported in 4-Gig memory platforms.
5. The configuration replace feature does not work if the FEX line card is offline.
6. The configuration replace feature is not supported on Cisco Nexus 9500 platform switches with -R line cards.
7. The configuration replace feature does not support autoconfigurations.

Supported Features

1. The configuration replace feature is supported on Cisco Nexus 3000 Series and Cisco Nexus 9000 Series switches.
2. Using the **configure replace** command from maintenance mode with a *<non-interactive>* option is supported. It takes the *yes* user-confirmation by default and proceeds.
3. The configuration replace feature is supported only for the configure terminal mode commands. The configure profile, configure jobs, and any other modes are not supported.
4. Beginning with Cisco NX-OS Release 9.3(4), the configuration replace feature is supported for breakout interface configurations.

Recommended Guidelines

1. We do not recommend a configuration replace operation that is performed on the configuration file that is generated across the software version because this operation could fail. A new configuration file must be regenerated whenever there is change in the software version.
2. We recommend that you do not change any configuration from others sessions if the configuration replace operation is in progress because it could cause the operation to fail.

Configuration Guidelines

1. When different versions of a feature are present in the running configuration and user configuration (for example: VRRPv2 and VRRPv3), semantic validation option does not work as expected. This issue is a known limitation.
2. In the verify-only mode, the TCAM-dependent configuration may not throw an error and gets succeeded. However, it may fail during actual CR operation. To avoid this, it is recommended to apply TCAM carving configuration and reload before performing CR.
3. Replay of configuration generated from one device and applied on another device goes through and CLI CR is successful but SNMPv3 operations fail with 'Authentication failure' message in the new device.
4. The user provided configuration file must be the valid show running-configuration output that is taken from the Cisco NX-OS device (copy run file). The configuration cannot be a partial configuration and must include mandated commands, such as user admin and so on.

5. The order of the commands in the user provided configuration file must be the same as those commands in the running configuration of the Cisco Nexus switch.
6. The user configuration file to which you need to replace the running configuration on the switch using CR should be generated from the running-config of the switch after configuring the new commands. The user configuration file should not be manually edited with the CLI commands and the sequence of the configuration commands should not be altered.
7. When **system default switchport shutdown** or **no system default switchport shutdown** is used with **configure replace bootflash:target_config_file** command, the user should make sure that desired port state (shutdown or no shutdown) statement is present in the target_config_file for all switchport interfaces.
8. Clear text passwords are allowed in the case of configuration replace candidate-config file as the user database gets synced between SNMP and AAA (Security).

Release Specific Guidelines

- Beginning with Cisco NX-OS Release 9.3(5) the configuration replace feature is supported on port profiles. The configure jobs mode is supported. Configuration files with scheduler job commands can be used for configuration replace. Configuration replace (CR) for FEX interface configurations is supported. Provisioning of FEX is not supported through CR. Once provisioned, configurations on the FEX interfaces can modified through CR.
- Beginning with Cisco NX-OS Release 9.3(6), the **boot nxos** *image* configuration can be excluded in the **show running-config** , **show startup-config** , **copy running-config** *filename* , and **copy startup-config** *filename* commands by configuring **service exclude-bootconfig** .
- Beginning with NX-OS Release 10.3(1)F, the configuration replace feature does not support feature app-hosting.
- Beginning from Cisco NX-OS Release 10.4(2)F, the configuration replace feature is supported for LDAP on Cisco NX-OS devices.
- Beginning from Cisco NX-OS Release 10.4(2)F, for non-case sensitive commands, if there is a letter case distinction between the commands in running config and candidate-config files, then the output of **config replace show-patch** displays both the commands due to the difference in letter case.
- Beginning from Cisco NX-OS Release 10.4(3)F, you can also use polymorphic commands in candidate configuration to perform configuration replace.
- Beginning with Cisco NX-OS Release 10.5(1)F, the following cannot be part of the same CR candidate file:
  - **no hardware access-list update atomic**
  - ACL configuration that exceeds existing running-configuration atomic TCAM configuration limits
- Beginning with Cisco NX-OS Release 10.5(1)F, sequence number is mandatory for vlan access-map command. Without a sequence number the configuration replace operation fails.

Perform Configuration Replace

Replace the running configuration on the switch with a candidate or user-provided configuration file using the **configure replace** command.

To perform configuration replace, do the following:

Before you begin

Ensure that there is no conflict in ip address in the current configuration and candidate configuration files. An example for conflict in ip address is—consider that you configured 172.16.0.1/24 on eth interface 1/53 on the current configuration file and port channel 30 with 172.16.0.1/24 and 192.168.0.1/24 on eth 1/53 in the candidate configuration file. When you perform a configuration replace to the candidate configuration file, this results in a conflict in ip address.

Procedure

Step 1 Perform configuration replace using the command **configure replace** { < *uri_local* > | < *uri_remote* > } [**verbose** | **show-patch**]

If you make the configuration changes through any sessions when configuration replace is in progress, the configuration replace operation fails. If you send a configuration replace request when one configuration request is already in progress, then it gets serialized.

Step 2 Display the differences between the running-configuration and the user-provided configuration using the command **configure replace** [**bootflash** / **scp** / **sftp**] < *user-configuration-file* > **show-patch**

Note
- This command does not encrypt plain text password.
- This command can still show a patch even after configuration replace is successful for CLI snmp-server traps commands .

Step 3 Replace the configuration on the switch with the new user configuration that is provided by the user using the command **configure replace** [**bootflash** / **scp** / **sftp**] < *user-configuration-file* > **verbose**

Configuration replace is always atomic.

Step 4 Replace the configuration on the switch with the new user configuration and enables the configuration replace with semantic validation using the command **configure replace** <*user-configuration-file*> [**best-effort**]

The best-effort option enables the configuration replace to execute the full patch despite any error in the commands and also make sure that the previous configuration is not rolled back.

Beginning with Cisco NX-OS Release 10.5(1)F, configuration replace feature supports batch ACL configurations on Cisco Nexus 9300-FX2/FX3/GX Series switches. If the **best effort** mode is enabled, any failure within the batched configuration will result in skipping the entire set of configurations in that particular batch.

Step 5 Replace the configuration on the switch with the new user configuration and enables the configuration replace with semantic validation using the command **configure replace** <*user-configuration-file*> [**verify-and-commit**]

The verify-and-commit option is used for enabling the semantic validation. Patch will be executed only if semantic validation of the full patch gets passed.

You can use the best-effort option or the verify-and-commit option or both the options at the same time.

Step 6 Show only the patch and does Semantic validation on the patch, and display the results using the command **configure replace** <*user-configuration-file*> [**verify-only**]

The patch does not get applied to the system.

Step 7 Configure the commit time in seconds using the command **configure replace** [**bootflash** / **scp** / **sftp**] < *user-configuration-file* > **verbose commit-timeout** <*time*>

This is an optional step. The timer starts after the configuration replace operation is successfully completed.

Step 8 Stop the commit timer and continues the configuration replace configuration using the command **configure replace** [**commit**] and use the command **configure replace** [**bootflash** / **scp** / **sftp**] <*user-configuration-file*> *non-interactive* for maintainence mode.

This is an optional step.

Note This step is applicable only if you have configured the commit-timeout feature.

Note To rollback to the previous configuration, you must wait for the expiry of the commit timer. Once the timer expires, the switch is automatically rolled back to the previous configuration.

There is no user prompt in maintenance mode. The **yes** user-confirmation is taken by default, and rollback proceeds. You can use the non-interactive option only in the maintenance mode.

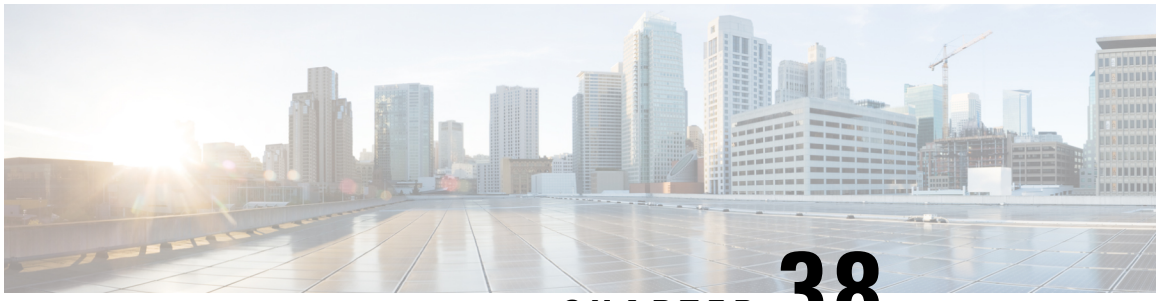
Step 9 (Optional) Reload the switch to the previously saved startup configuration if configure replace operation fails and the restore to the previous running configuration also fails using the command **configure replace** <*user-configuration-file*> [**on-failure reload** [**save-startup-config**]]

This is an optional step.

This option has no impact if:

- Configure replace is successful
- Configure replace fails, but restore to the previous running configuration is successful

Note [**save-startup-config**] – This option saves the running config to startup before starting the configure replace operations.



CHAPTER 38

Rollback

A rollback allows you to take a snapshot, or user checkpoint, of the Cisco NX-OS configuration and then reapply that configuration to your device at any point without having to reload the device. A rollback allows any authorized administrator to apply this checkpoint configuration without requiring expert knowledge of the features configured in the checkpoint.

- Overview, on page 625
- Prerequisites for Rollbacks, on page 626
- Guidelines and Limitations, on page 627
- Configuration, on page 628
- Create Checkpoint, on page 628
- Implement Rollback, on page 629
- Additional References, on page 630

Overview

Cisco NX-OS automatically creates system checkpoints. You can use either a user or system checkpoint to perform a rollback.

You can create a checkpoint copy of the current running configuration at any time. Cisco NX-OS saves this checkpoint as an ASCII file which you can use to roll back the running configuration to the checkpoint configuration at a future time. You can create multiple checkpoints to save different versions of your running configuration.

When you roll back the running configuration, you can trigger the following rollback types:

- atomic—Implement a rollback only if no errors occur.
- best-effort—Implement a rollback and skip any errors.
- stop-at-first-failure—Implement a rollback that stops if an error occurs.

The default rollback type is atomic.

When you are ready to roll back to a checkpoint configuration, you can view the changes that will be applied to your current running configuration before committing to the rollback operation. If an error occurs during the rollback operation, you can choose to cancel the operation, or ignore the error and proceed with the rollback. If you cancel the operation, Cisco NX-OS provides a list of changes already applied before the error occurred. You need to clean up these changes manually.

Supported Features

- High Availability -

  Whenever a checkpoint is created using the checkpoint or checkpoint checkpoint_name commands, the checkpoint is synchronized to the standby unit.

  A rollback remembers the states of the checkpoint operation, so if the checkpoint operation is interrupted and the system is left in an inconsistent state, a rollback can complete the checkpoint operation (synchronize the checkpoint with the standby unit) before proceeding with the rollback operation.

  Your checkpoint files are still available after a process restart or supervisor switchover. Even if there is an interruption during the process restart or supervisor switchover, the checkpoint will complete successfully before proceeding with the operation. In a supervisor switchover, the checkpoint is completed on the new active unit.

  If a process restart or supervisor switchover occurs during a rollback operation, after the restart or switchover completes, the rollback will resume from its previous state and complete successfully.

- Virtualization -

  Cisco NX-OS creates a checkpoint of the running configuration. You can create different checkpoint copies.

Automatically Generated System Checkpoints

The Cisco NX-OS software automatically generates system checkpoints to help you avoid a loss of configuration information. System checkpoints are generated by the following events:

- Disabling an enabled feature with the **no feature** command
- Removing an instance of a Layer 3 protocol, such as with the **no router bgp** command or the **no ip pim sparse-mode** command
- License expiration of a feature

If one of these events causes system configuration changes, the feature software creates a system checkpoint that you can use to roll back to the previous system configuration. The system generated checkpoint filenames begin with “system-” and include the feature name. For example, the first time that you disable the EIGRP feature, the system creates the checkpoint named system-fm-__inst_1__eigrp.

Default Settings

This table lists the default settings for rollback parameters.

Parameters	Default
Rollback type	Atomic

Prerequisites for Rollbacks

- To configure rollback, you must have network-admin user privileges.

- After you run **reload ascii**, run **copy running-config startup-config** before you create a checkpoint, even if the running and startup configurations appear identical. Also run **copy running-config startup-config** after **write erase** and **reload** before you create a checkpoint.

Guidelines and Limitations

- Only one user can perform a checkpoint, rollback, or copy the running configuration to the startup configuration at the same time.
- After the system executes the **write erase** and **reload** command, checkpoints are deleted. You can use the **clear checkpoint database** command to clear out all checkpoint files.
- Although a rollback is not supported for checkpoints across software versions, users can perform a rollback at their own discretion and can use the best-effort mode to recover from errors.
- When checkpoints are created on bootflash, differences with the running-system configuration cannot be performed before performing the rollback, and the system reports “No Changes.”
- Checkpoints created using the **checkpoint** and **checkpoint** *checkpoint_name* commands are present upon a switchover.
- Checkpoints are present upon reload unless a **write-erase** command is issued before a reload.
- A rollback to files on bootflash is supported only on files that are created using the **checkpoint** *checkpoint_name* command and not on any other type of ASCII file.
- Rollback is not supported in the context of auto configurations. Checkpoints do not store auto configurations. Therefore, after a rollback is performed, the corresponding auto configurations will not be present.
- Multiple port VLAN mappings configured on an interface during a rollback operation cause the rollback feature to fail.

Checkpoint Filename Guidelines



Note To configure rollback, you must have network-admin user privileges.

- You can create up to ten checkpoint copies.
- Your checkpoint filenames must be 80 characters or less.
- You cannot start a checkpoint filename with the word *system* .
- You can start a checkpoint filename with the word *auto* .
- You can name a checkpoint file *summary* or any abbreviation of the word *summary* .
- Checkpoint names must be unique. You cannot overwrite previously saved checkpoints with the same name.

Configuration

Verify Configuration



Note Be aware that the Cisco NX-OS commands may differ from the Cisco IOS commands.

To display the rollback configuration information, perform one of the following tasks:

Command	Purpose
show checkpoint *name* [**all**]	Displays the contents of the checkpoint name.
show checkpoint all [**user** | **system**]	Displays the contents of all checkpoints. You can limit the displayed checkpoints to user or system generated checkpoints.
show checkpoint summary [**user** | **system**]	Displays a list of all checkpoints. You can limit the displayed checkpoints to user or system generated checkpoints.
show diff rollback-patch {**checkpoint** *src-cp-name* | **running-config** | **startup-config** | **file** *source-file*} {**checkpoint** *dest-cp-name* | **running-config** | **startup-config** | **file** *dest-file*}	Displays the differences between the source and destination checkpoint selections.
show rollback log [**exec** | **verify**]	Displays the contents of the rollback log.

Use the **clear checkpoint database** command to delete all checkpoint files.

Configuration Example

This example shows how to create a checkpoint file and then implements a best-effort rollback to a user checkpoint name:

```
checkpoint stable
rollback running-config checkpoint stable best-effort
```

Create Checkpoint

You can create up to ten checkpoints of your configuration.

Procedure

Step 1 Create a checkpoint of the running configuration to either a user checkpoint name or a file using the command [**no**] **checkpoint** {[*cp-name*] [**description** *descr*] | **file** *file-name* }

Example:

```
switch# checkpoint stable
```

The checkpoint name can be any alphanumeric string up to 80 characters but cannot contain spaces. If you do not provide a name, Cisco NX-OS sets the checkpoint name to user-checkpoint- *number* where *number* is from 1 to 10.

The description can contain up to 80 alphanumeric characters, including spaces.

You can use the **no** form of the **checkpoint** command to remove a checkpoint name. Use the **delete** command to remove a checkpoint file.

Step 2 Display the contents of the checkpoint name using the command **show checkpoint** *cp-name* [**all**]

Example:

```
switch# show checkpoint stable
```

This is an optional step.

Implement Rollback

You can implement a rollback to a checkpoint name or file. Before you implement a rollback, you can view the differences between source and destination checkpoints that reference current or saved configurations.

Note If you make a configuration change during an atomic rollback, the rollback will fail.

Procedure

Step 1 Display the differences between the source and destination checkpoint selections using the command **show diff rollback-patch** { **checkpoint** *src-cp-name* | **running-config** | **startup-config** | **file** *source-file* } { **checkpoint** *dest-cp-name* | **running-config** | **startup-config** | **file** *dest-file* }

Example:

```
switch# show diff rollback-patch checkpoint stable running-config
```

Step 2 Create a rollback to the specified checkpoint name or file using the command **rollback running-config** { **checkpoint** *cp-name* | **file** *cp-file* } [**atomic** | **best-effort** | **stop-at-first-failure**]

Example:

```
switch# rollback running-config checkpoint stable
```

You can implement the following rollback types:

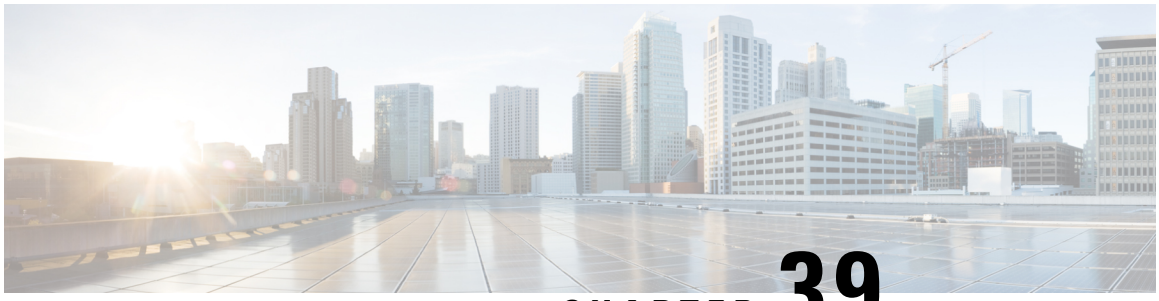
- **atomic** —Implement a rollback only if no errors occur.
- **best-effort** —Implement a rollback and skip any errors.
- **stop-at-first-failure** —Implement a rollback that stops if an error occurs.

The default is atomic.

This example shows how to implement a rollback to a user checkpoint name.

Additional References

Configuration files - *Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide*



CHAPTER 39

Candidate Configuration

- Overview, on page 631
- Guidelines and Limitations, on page 633
- Perform Integrity Check for Candidate Config, on page 638
- Configuration Candidate session, on page 639

Overview

Candidate configuration is a subset of the running-config which checks whether the Candidate config exists in the running-config without any additions or modifications or deletions.

To check the integrity of the candidate config, use the following commands:

- `show diff running-config`
- `show diff startup-config`

For more information on the CLIs, refer to Perform Integrity Check for Candidate Config, on page 638.

Examples

No Difference Between Running and Candidate Config

```
switch# show diff running-config bootflash:base_running.cfg
switch#
```

Difference Between Running and Candidate

```
switch# show diff running-config bootflash:modified-running.cfg unified
--- running-config
+++ User-config
@@ -32,11 +32,11 @@

interface Ethernet1/1
   mtu 9100
   link debounce time 0
   beacon
-  ip address 2.2.2.2/24
+  ip address 1.1.1.1/24
   no shutdown

interface Ethernet1/2
```

```
interface Ethernet1/3
switch#
```

Difference Between Running and Partial Candidate

```
switch# show file bootflash:intf_vlan.cfg
interface Vlan101
  no shutdown
  no ip redirects
  ip address 1.1.2.1/24 secondary
  ip address 1.1.1.1/24
switch#
switch# show diff running-config bootflash:intf_vlan.cfg partial unified
--- running-config
+++ User-config
@@ -3897,10 +3883,14 @@
   mtu 9100
   ip access-group IPV4_EDGE in
   ip address 2.2.2.12/26 tag 54321

 interface Vlan101
+ no shutdown
+ no ip redirects
+ ip address 1.1.2.1/24 secondary
+ ip address 1.1.1.1/24

 interface Vlan102
   description Vlan102
   no shutdown
   mtu 9100
switch#
```

Partial Configuration Diff Merged

```
switch# show file po.cfg
interface port-channel500
description po-123
switch#
switch# sh run int po500

!Command: show running-config interface port-channel500
!Running configuration last done at: Fri Sep 29 12:27:28 2023
!Time: Fri Sep 29 12:30:24 2023

version 10.4(2) Bios:version 07.69

interface port-channel500
  ip address 192.0.2.0/24
  ipv6 address 2001:DB8:0:ABCD::1/48

switch#

switch# show diff running-config po.cfg partial merged unified
--- running-config
+++ User-config
@@ -124,10 +110,11 @@
interface port-channel100
 interface port-channel500
   ip address 192.0.2.0/24
   ipv6 address 2001:DB8:0:ABCD::1/48
+ description po-123
 interface port-channel4096
 interface Ethernet1/1
switch#
```

Guidelines and Limitations

- The line numbers that are displayed in the generated running config do not match with the candidate config as they are internally generated one.
- If there is any difference between the configuration of running and candidate, then it is displayed inline as output.
- If the whole block of configuration in the candidate file is a new addition, it will be appended at the end of the generated running config.
- When the candidate config has an SNMP or an AAA user CLI with clear-text password, the SNMP user is seen as a diff even when the user is already configured.
- EIGRP address family IPv4 configs are recommended to configure under the EIGRP address family hierarchy and not under the router mode hierarchy in the candidate file, before running a partial diff.
- If the target/candidate file has a default command (for example, `- log-neighbor-warnings;`) configured directly under the **router eigrp** mode and not one of its submodes, that is, **address-family ipv4 unicast** or **address-family ipv6 unicast** , then partial-diff shows + displayed in the output of the default command (for example, `+ log-neighbor-warnings`) in the diff.
- For noncase sensitive commands, if there is a letter case distinction between the commands in the running config and candidate-config files, then the output of **partial diff** displays both the commands due to the difference in letter case.
- Cleartext passwords are allowed in case of partial diff candidate CONFIG_FILE as the user database gets synced between SNMP and AAA (Security).
- Configuration profile, maintenance profile (mmode) and scheduler mode configurations are not supported.

Integrity Check Guidelines

1. To perform an integrity check on a full running configuration file, the user must supply a running configuration generated by an NX-OS system and should not use the **partial** keyword. The **partial** keyword treats the input candidate as a subset of the running configuration.

Release Specific Guidelines

1. Beginning with Cisco NX-OS Release 10.2(3)F, Candidate config integrity check option is introduced on all Cisco Nexus switches.
2. Beginning from Cisco NX-OS Release 10.4(3)F, you can also use polymorphic commands in candidate configuration to perform partial diff.

Partial Diff of Default Commands for Multicast Components Guidelines

The content of this section is applicable from Cisco NX-OS Release 10.4(3)F.

If the default commands of multicast components are present in the candidate CONFIG_FILE, they are seen in show diff as follows:

Multicast Component	Default Commands in show diff
PIM	ip access-list copp-system-p-acl-pim 10 permit pim any 224.0.0.0/24 20 permit udp any any eq pim-auto-rp ip access-list copp-system-p-acl-pim-mdt-join ip access-list copp-system-p-acl-pim-reg 10 permit pim any any
PIM6	ipv6 access-list copp-system-p-acl-pim6 10 permit pim any ff02::d/128 20 permit udp any any eq pim-auto-rp ipv6 access-list copp-system-p-acl-pim6-reg 10 permit pim any any
IGMP	ip access-list copp-system-p-acl-igmp 10 permit igmp any 224.0.0.0/3 class-map copp-system-p-class-normal-igmp
MLD	ipv6 access-list copp-system-p-acl-mld 10 permit icmp any any mld-query 20 permit icmp any any mld-report 30 permit icmp any any mld-reduction 40 permit icmp any any mldv2

show diff running-config *file_url* [unified] [partial] [merged] Command Guidelines

- When using the **unified** , **partial** , and **merged** option to review the differences for the following PBR commands, the diff outputs are as mentioned below:
  - **set ip next-hop**
  - **set ip default next-hop**
  - **set ip default vrf next-hop**
  - **set ipv6 next-hop**
  - **set ipv6 default next-hop**
  - **set ipv6 default vrf next-hop**

1. If the candidate next-hops are a subset of running next-hops (in the same order and sequence), and candidate additive flags are a subset of running flags, then the diff output is empty as shown in the following table:

Candidate Config	Running Config	Partial Unified Merged Diff Output
route-map rmap1 permit 10 set ip next-hop 1.1.1.1 2.2.2.2 load-share	route-map rmap1 permit 10 set ip next-hop 1.1.1.1 2.2.2.2 3.3.3.3 load-share force-order	<no-diff>

2. If the candidate next-hops are a subset of running next-hops (in the same order and sequence), and the candidate has some extra additive flags which are not present in running config, then the diff output appends any additional flags present in the candidate config to the running config, similar to command line behavior as shown in the following table:

Candidate Config	Running Config	Partial Unified Merged Diff Output
route-map rmap1 permit 10 set ip next-hop 1.1.1.1 2.2.2.2 load-share force-order	route-map rmap1 permit 10 set ip next-hop 1.1.1.1 2.2.2.2 3.3.3.3 load-share drop-on-fail	route-map rmap1 permit 10 - set ip next-hop 1.1.1.1 2.2.2.2 3.3.3.3 load-share drop-on-fail + set ip next-hop 1.1.1.1 2.2.2.2 3.3.3.3 load-share force-order drop-on-fail

3. If candidate next-hops are not a subset of running next-hops (in the same order and sequence), and the candidate and running record can have any additive flag, then the diff output indicates this with a '–' for the running config record and a '+' for the candidate config record.

   This distinction is important, particularly when using with PBR commands, where the sequence of next-hops is critical. Even if the next-hops IP addresses are identical, their order affects functionality.

   For example, '1.1.1.1 2.2.2.2' is different from '2.2.2.2 1.1.1.1'.



Important If there is an additive flag in the running config that you wish to retain after merging with the candidate config, you must explicitly include that flag in the candidate config. This ensures that the needed flags are preserved in the final, merged configuration.

Candidate Config	Running Config	Partial Unified Merged Diff Output
route-map rmap1 permit 10 set ip next-hop 1.1.1.1 2.2.2.2 load-share drop-on-fail	route-map rmap1 permit 10 set ip next-hop 2.2.2.2 1.1.1.1 load-share force-order	route-map rmap1 permit 10 - set ip next-hop 2.2.2.2 1.1.1.1 load-share force-order + set ip next-hop 1.1.1.1 2.2.2.2 load-share drop-on-fail

- When **Partial Unified** or **Partial Unified Merged** option is used, all the PBR commands are mutually exclusive and cannot coexist within the same parent route-map. Therefore, if a candidate configuration specifies multiple mutually exclusive PBR commands under a single route-map, only the last command variant will be shown in the partial diff output.

  Example-1: In this example, the candidate configuration includes multiple PBR commands under a single route-map **rmap1** :

```
route-map rmap1 permit 10
  set ip next-hop 1.1.1.1 2.2.2.2
  set ipv6 next-hop 3::3
  set ip next-hop verify-availability 4.4.4.4
  set ip next-hop verify-availability 5.5.5.5
  set ip vrf green next-hop 6.6.6.6
  set ip vrf blue next-hop 7.7.7.7 8.8.8.8
```

  Before the generation of the partial-diff output, the above candidate configuration is automatically converted to the following:

```
route-map rmap1 permit 10
  set ip vrf green next-hop 6.6.6.6
  set ip vrf blue next-hop 7.7.7.7 8.8.8.8
```

Example-2: In this example, the candidate configuration includes multiple 'set ip next-hop verify-availability' commands with different track IDs specified for the route-map **rmap2** . Since track IDs cannot be modified for the same next-hop, these commands are mutually exclusive:

```
route-map rmap2 permit 10
  set ip next-hop verify-availability 1.1.1.1 track 1
  set ip next-hop verify-availability 2.2.2.2 track 20
  set ip next-hop verify-availability 2.2.2.2 track 30
  set ip next-hop verify-availability 2.2.2.2 track 40
  set ip next-hop verify-availability 3.3.3.3 track 3
```

Before generating the partial-diff output, the system will automatically consolidate these commands by retaining only the last **set ip next-hop verify-availability** command for each next-hop IP address as shown below:

```
                                route-map rmap2 permit 10
  set ip next-hop verify-availability 1.1.1.1 track 1
                                set ip next-hop verify-availability 2.2.2.2 track 40
  set ip next-hop verify-availability 3.3.3.3 track 3
```

- When the **Partial Unified Merged** option is used, to review the differences for the **verify-availability** command variants, the track ID for a given next-hop is not modifiable.

  Therefore, if the candidate and running configurations contain the same next-hop but have different track IDs under the same parent route-map, the candidate record cannot simply be merged with the running record, as per command line behavior. Therefore, to apply the candidate record with different track ID for the same next-hop, the corresponding running config record must be removed first ('–' for the running configuration record in the diff) and then when the candidate record is merged, it will be appended at the end of the last record under the same parent route-map ('+' for the candidate config record).

  The following table shows the sample candidate and running configuration with the 'Partial Unified Merged' output for different use cases as mentioned below:

  1. If the track ID is different for the same next-hop under candidate and running config, then the diff output is as shown in the following table:

Candidate Config	Running Config	Partial Unified Merged Diff Output

route-map rmap1 permit 10
  set ip next-hop
verify-availability 1.1.1.1
 track 1
  set ip next-hop
verify-availability 2.2.2.2
 track 20
  set ip next-hop
verify-availability 3.3.3.3
 track 3 load-share
	route-map rmap1 permit 10
	  set ip next-hop
	verify-availability 1.1.1.1
	 track 1
	  set ip next-hop
	verify-availability 2.2.2.2
	 track 2
	  set ip next-hop
	verify-availability 3.3.3.3
	 track 3 load-share
		route-map test permit 10
		  set ip next-hop
		verify-availability 1.1.1.1
		track 1
		 - set ip next-hop
		verify-availability 2.2.2.2
		track 2
		  set ip next-hop
		verify-availability 3.3.3.3
		track 3
		 + set ip next-hop
		verify-availability 2.2.2.2
		track 20 load-share

  2. If track ID is not present under candidate config but present in running config for the same next-hop, then the diff output is empty as shown in the following table:

Candidate Config	Running Config	Partial Unified Merged Diff Output

route-map rmap1 permit 10
  set ip next-hop
verify-availability 1.1.1.1
 track 1
  set ip next-hop
verify-availability 2.2.2.2
 	
  set ip next-hop
verify-availability 3.3.3.3
 track 3
	route-map rmap1 permit 10
	  set ip next-hop
	verify-availability 1.1.1.1
	 track 1
	  set ip next-hop
	verify-availability 2.2.2.2
	 track 2
	  set ip next-hop
	verify-availability 3.3.3.3
	 track 3
		no-diff

3. If track ID is not present under running config but present in candidate config for the same next-hop, then the diff output is as shown in the following table:

Candidate Config	Running Config	Partial Unified Merged Diff Output

route-map rmap1 permit 10
  set ip next-hop
verify-availability 1.1.1.1
 track 1
  set ip next-hop
verify-availability 2.2.2.2
 track 20
  set ip next-hop
verify-availability 3.3.3.3
 track 3
	route-map rmap1 permit 10
	  set ip next-hop
	verify-availability 1.1.1.1
	 track 1
	  set ip next-hop
	verify-availability 2.2.2.2
	
	  set ip next-hop
	verify-availability 3.3.3.3
	 track 3
		route-map rmap1 permit 10
		  set ip next-hop
		verify-availability 1.1.1.1
		track 1
		 - set ip next-hop
		verify-availability 2.2.2.2
		  set ip next-hop
		verify-availability 3.3.3.3
		track 3
		 + set ip next-hop
		verify-availability 2.2.2.2
		track 20

Partial Diff of RPM Commands Guidelines

The content of this section is applicable from Cisco NX-OS Release 10.4(3)F.

When using the unified, partial, and merged option to review the differences for the following RPM commands, the diff outputs are as follows:

- In the candidate configuration, the RPM commands will undergo syntactic validation as reflected in the diff output. However, semantic validation will not be performed in the diff output. It is the user's responsibility to ensure that the commands in the candidate configuration are semantically accurate.

  If the command in the Candidate-config is semantically incorrect, the diff may incorrectly indicate that the command is executable, but in actual it may not.

- Ensure that you provide the sequence number mandatorily for the following commands in the Candidate-config file:
  - **ip prefix-list list-name seq** *seq* { **deny** | **permit** } *prefix*
  - **ipv6 prefix-list list-name seq** *seq* { **deny** | **permit** } *prefix*
  - **mac-list list-name seq** *seq* { **deny** | **permit** } *prefix*
  - **ip community-list** { **standard** | **expanded** } *list-name* seq *seq* { **deny** | **permit** } *expression*
  - **ip extcommunity-list** { **standard** | **expanded** } *list-name* seq *seq* { **deny** | **permit** } *expression*
  - **ip large-community-list** { **standard** | **expanded** } *list-name* seq *seq* { **deny** | **permit** } *expression*

- **ip-as-path access-list** *list-name* seq *seq* { **deny** | **permit** } *expression*

- When the following commands include an expression string that has spaces enclosed in quotes within the Candidate-config, there will be no differences displayed in the diff output:
  - **ip community-list expanded** *list-name* seq *seq* { **deny** | **permit** } *expression*
  - **ip extcommunity-list expanded** *list-name* seq *seq* { **deny** | **permit** } *expression*
  - **ip large-community-list expanded** *list-name* seq *seq* { **deny** | **permit** } *expression*
  - **ip-as-path access-list** *list-name* seq *seq* { **deny** | **permit** } *expression*

Candidate Config	Running Config	Partial Unified [Merged] Diff Output
`ip community-list expanded list_abc seq 10 permit "1:1 "`	`ip community-list expanded list_abc seq 10 permit "1:1"`	`no-diff`
`ip extcommunity-list expanded list_abc seq 10 permit "1:1  "`	`ip extcommunity-list expanded list_abc seq 10 permit "1:1"`	`no-diff`
`ip large-community-list expanded list_abc seq 10 permit "1:1:1   "`	`ip large-community-list expanded list_abc seq 10 permit "1:1:1"`	`no-diff`
`ip as-path access-list list_abc seq 10 permit "1 "`	`ip as-path access-list list_abc seq 10 permit "1"`	`no-diff`

Perform Integrity Check for Candidate Config

To perform the integrity check, use the following commands:

Before you begin

Note Before performing the integrity check, ensure that the running config and the candidate config belong to the same image version.

Procedure

Step 1 Display the differences between the running and user given candidate config using the command **show diff running-config** *file_url* [**unified**] [**merged**]

Example:

```
switch# show diff running-config bootflash:candidate.cfg partial unified
```

- *file_url* : File path to compare with.

- **unified** : Displays the difference between running and user configuration in unified format.
- **merged** : Enter **merged** only if sub-commands need to be merged instead of replace.

Step 2 Display the differences between the startup and user given candidate config using the command **show diff startup-config** *file_url* [**unified**]

Example:

```
switch# show diff startup-config bootflash:candidate.cfg unified
```

- *file_url* : File path to compare with.
- **unified** : Displays the difference between startup and user configuration in unified format.

Configuration Candidate session

A Configuration Candidate session is a configuration management feature that allows you to stage, review, and validate configuration changes before committing them to the running configuration on a Cisco Nexus switch.

- Staged configuration changes are isolated from the active running configuration until you explicitly commit them.
- You can compare staged changes to the current running configuration to review differences and validate intended updates.
- Candidate session supports committing all changes at once, discarding staged changes, and rolling back to previous commit points.
- Candidate session can be initiated for both manual CLI-based changes and automated workflows using supported programming interfaces.
- Sessions are not persistent across switch reloads; any uncommitted changes are lost if the switch is restarted.
- Only a limited number of Candidate sessions can exist at one time, and only one user can access a session at a time.

Candidate session Use Cases and Administrator Value

- Prepare and review configuration changes in advance, minimizing the risk of unintended disruptions.
- Validate changes by viewing differences before applying them, which helps identify configuration errors or conflicts.
- Apply all intended changes at once, or abort the session to discard them without affecting the active configuration.
- Integrate with automation and programmatic interfaces for batch configuration workflows.

Actors in the Process

- Network administrator: Initiates and manages the configuration session.
- Switch CLI infrastructure: Provides the commands and session environment for staging, reviewing, and committing changes.
- Data Management Engine (DME): Maintains the candidate configuration database for staging changes.

In summary, Candidate session allows you to isolate changes from the active configuration, review differences, validate updates, and then commit or discard them as a single unit.

Guidelines and Limitations for Configuration Candidate Session

Beginning with NX-OS Release 10.6(3)F, candidate session (Configuration Staging) is supported on Cisco Nexus 9000 switches. Candidate sessions allow you to stage and review configuration changes before committing them to the running configuration.

- Creation of a new configuration candidate session is blocked if the current running configuration contains unsupported commands. The error message displayed when a command is not supported is as follows:

  ```
  switch# configure candidate
  Error: Current running-config has commands that are not supported by the candidate
  session.
  Please check 'show candidate-config last-unsupported' and remove the unsupported
  configurations from the running-config.
  switch#
  ```

  To resolve this issue, run the following command to identify unsupported configurations:

  ```
  switch# show candidate-config last-unsupported
  ```

  Then remove the unsupported commands from the running configuration before creating a new candidate session.

- Candidate session is removed if the device reloads or the supervisor switches.
- Only one candidate session can be active.
- Only one user can access and modify each candidate session at a time.
- The system maintains a history of up to 50 commit entries; older entries are overwritten as new commits are made.
- A candidate session supports a maximum configuration size of 35,000 lines.
- Interface breakout and the associated configuration of the newly created interfaces cannot be performed in the same candidate session. Use two separate candidate sessions: first to perform the interface breakout, and second to apply the required interface configurations
- A candidate session name can contain up to 64 alphanumeric characters, hyphens (-), and underscores (_). Spaces and other special characters are not supported. Names cannot start with special characters

Configuring a Candidate Session

This task describes how to start a configuration session, stage configuration changes, review the changes, and commit or discard them.

Use candidate session to safely stage multiple configuration changes before committing them to the device.

- You can review your staged changes and compare them with the running configuration before applying them.
- Candidate session helps prevent accidental changes by allowing you to review and validate before commit.

Before you begin

- Device running NX-OS Release 10.6(3)F or later.
- RBAC permissions for candidate session creation and commit.
- Only one candidate session can be active on the device.

Procedure

Step 1 Start a configuration session and copy configuration from file to the candidate session.

Use the following command to start a candidate session:

```
switch#configure candidate [name <session-name>]
```

You can provide a session name, but it is optional. If you do not specify one, the system automatically generates a name in the format session-x.

Use the following command to copy configuration from file to candidate session:

```
switch(config-c-<session-name>)# import configuration <uri0>
```

Step 2 Stage configuration changes, review and compare them in the session.

Use the following command to stage configuration changes in the session

```
switch(config-c-<session-name>)# interface Ethernet1/1
```

The session accepts supported configuration commands. Unsupported commands are rejected with an error.

Use the following command to review staged changes and compare them with the running configuration:

```
switch(config-c-<session-name>)# show candidate-config diff
```

Use the diff command to see what changes will be made when the session is committed.

Step 3 Use any of the following commands to commit the staged changes.

- `switch(config-c-<session-name>)# commit [timeout <seconds>]`
- `switch# configure candidate name <session-name> commit`

Committed changes are applied to the running configuration.

Step 4 To add a description to the commit, use this command.

```
switch(config-c-<session-name>)# commit description this is a test
```

The description can contain up to 128 characters. You can use alphanumeric characters, spaces, underscores (_), hyphens (-), and periods (.). Enter a multiword description without quotation marks. Quotation marks cause an invalid command error.

Step 5 Use any of the following commands to discard the candidate session without applying the changes and delete it.

- `switch(config-c-<session-name>)# abort`
- `switch# configure candidate name <session-name> abort`

Step 6 Confirm or abort the commit before a timeout.

Use the following command to confirm or abort the commit before a timeout:

`switch# configure candidate name <session-name> commit [confirm] | abort`

Note If you choose to abort, all staged changes are discarded and the session is deleted.

Step 7 Reset the session configuration to the default configuration.

Use the following command to reset the session configuration to the default configuration:

`switch(config-c-<session-name>)# factory-reset candidate`

Note If you are importing the complete running configuration into the session, factory-reset the existing candidate session before initiating the import procedure

Step 8 Clean up the old Commit IDs from the system.

Use the following command to clean up the old Commit IDs from the system:

`switch# clear configuration commits {diskspace <size> | oldest <num>}`

Step 9 Roll back to a previous commit using its ID.

Use the following command to roll back to a previous commit using its ID:

`switch# rollback configuration commit <id>`

This step is optional and can be used if you need to revert changes.

Step 10 Delete the candidate session

Use the following command to delete the candidate session:

`switch# no configure candidate name <session-name>`

Configuration changes are staged, reviewed, and either applied to or discarded from the device.

After committing or aborting the session, verify the device configuration as needed.

Verifying a Candidate Session

Verification ensures that staged changes are correct before you commit them and that committed changes are present in the running configuration.

Confirm that the candidate session contains the intended staged changes and that committed changes have taken effect.

Perform verification after staging configuration changes and again after committing them to the device.

Procedure

Step 1 Display the staged configuration for the current session using the `switch(config-c-<session-name>)# show candidate-config` command.

This command shows all commands that are currently staged in the configuration session

Step 2 Compare staged changes with the running configuration using the `switch(config-c-<session-name>)# show candidate-config diff` command.

This command displays the differences between the staged configuration and the running configuration

Step 3 Verify committed changes in the running configuration by using the `switch# show running-config` command.

Use this step after you commit the session to ensure the intended changes are present.

Step 4 View commit history for candidate session using the `switch# show configuration commit list` command.

This shows a list of commit IDs that you can use for audit or rollback.

Step 5 View commit information for a session name by using the `switch# show configuration commit { session <name> | changes session <name>}` command.

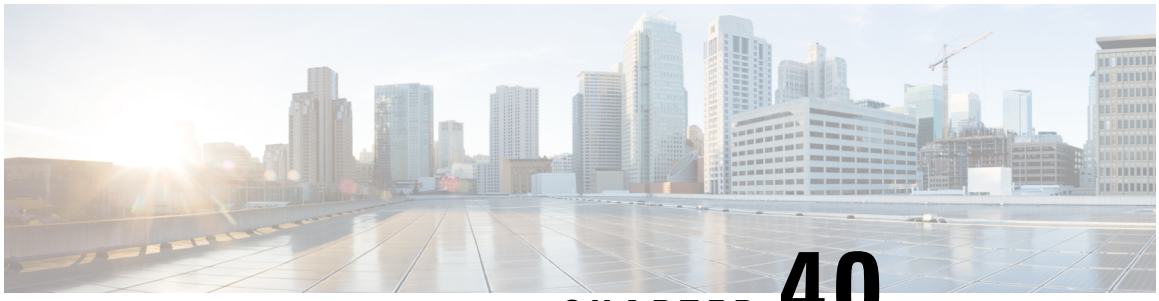
Step 6 View execution logs of last rollback by using the `switch# show configuration rollback log` command.

Step 7 View the commit ID info by using the `switch# show configuration commit {[changes] <commit-id>| list}` command.

Step 8 View the pending and completed candidate session by using the `switch# show candidate summary [pending | committed]` command.

Step 9 View the configuration of given candidate name by using the `switch# show candidate-config name <session-name>` command.

The candidate session is verified. You can confirm that staged changes match the intended configuration and that committed changes are present in the running configuration.



CHAPTER 40

Secure Erase

Cisco trustworthy technologies provide product assurance functionality as well as foundational security capabilities that enhance the security and resilience of Cisco solutions. These technologies are adopted by Cisco Nexus 9000 switches used in a data center. Secure Erase is one of the Cisco Trustworthy technologies.

- Secure erase, on page 645
- Guidelines and limitations for secure erase, on page 646
- Perform a secure erase, on page 646
- Secure erase examples, on page 648

Secure erase

Secure erase is a security feature that

- erases all customer information on Nexus 9000 switches,
- performs a factory reset that clears environment variables, and
- is useful while returning a device or recovering a compromised device.

Cisco Nexus 9000 switches consume storage to conserve system software images, switch configuration, software logs, and operational history. These areas can have customer-specific information such as details regarding network architecture, and design as well as a potential target for data thefts.

Beginning with Cisco NX-OS Release 10.2(2)F, the Secure Erase feature is introduced to erase all customer information for Nexus 9000 switches. Secure Erase is an operation to remove all the identifiable customer information on Cisco NX-OS devices in conditions of product removal due to Return Merchandise Authorization (RMA), or upgrade or replacement, or system end-of-life.

Note Secure erase feature does not erase content in external storage.

The secure erase process is used in the following two scenarios:

- Return Material Authorization (RMA) for a device - If you must return a device to Cisco for RMA, remove all the customer-specific data before obtaining an RMA certificate for the device.

- Recovering the compromised device - If the key material or credentials that are stored on a device is compromised, reset the device to factory configuration, and then reconfigure the device.

The device reloads to perform a factory reset which results in the EoR chassis modules to enter the power down mode. After a factory reset, the device clears all configuration, logs, and storage information.

Guidelines and limitations for secure erase

Ensure that you read the guidelines and limitations before performing secure erase.

FEX mode

- If you perform secure erase in FEX mode, the switch boots in TOR mode after completion of the secure erase operation.
- If you configure secure erase of a FEX, factory reset is initiated and FEX configuration is removed.
- Always monitor FEX secure erase using the FEX console. If failure occurs, reboot, bring up FEX, and initiate secure erase again.

Software patches

- Software patches installed on the device are not restored after the secure erase operation.

General guidelines and limitations

- If the **factory-reset** command is issued through a session, the session is not restored after the completion of the factory reset process.
- The top of rack switches and supervisor modules return to the loader prompt.
- End of row switch modules remain in a powered-down state.

Feature support

- Starting from 10.2(2)F, Secure Erase feature is introduced in NX-OS, and 9300-FX3, -FX3S, and -FX3P switches are supported in TOR and FEX mode.
- Beginning with Cisco NX-OS Release 10.4(1)F, Secure Erase is supported on N9K-C9332D-H2R switch.
- Beginning with Cisco NX-OS Release 10.4(2)F, Secure Erase is supported on N9K-C93400LD-H1 switch.
- Beginning with Cisco NX-OS Release 10.4(3)F, Secure Erase is supported on N9K-C9364C-H1 switch.

Perform a secure erase

Use the secure erase procedure to delete all the necessary data before shipping to RMA.

Note Parallel secure erase operations are not supported.

To erase multiple modules in a single EoR chassis, the recommended order is: line card, fabric, standby supervisor, system controller, and then active supervisor.

Before you begin

- Ensure that you back up all the software images, configurations, and personal data.
- Ensure that there is an uninterrupted power supply while performing secure erase.
- Ensure that In-Service Software Upgrade (ISSU) or In-Service Software Downgrade (ISSD) is not in progress.

Procedure

Use the **factory-reset module** *mod* command to perform a factory reset. You can also use the command with **all** option. To use the factory reset command, no system configuration is required.

Example:

```
switch(config)# factory-reset [module <3>]
```

To secure erase FEX, use the **factory-reset fex** [*all fex_no*] command. To secure erase all FEX at once, use the **all** option.

Note Ensure that the FEX is not in Active-Active scenario, before initiating secure erase operation.

Additional options

Use the option **mod** to reset the start-up configurations:

- For top of rack switches, use the **factory-reset** or **factory-reset module 1** command.
- For top of rack switches in LXC mode, use the **factory-reset module 1** or **27** command.
- For end of row module switches, use the [{ **module** < *module* > [**bypass-secure-erase**] [**preserve-image**]}] command.

  Beginning with Cisco NX-OS Release 10.2(3), the following options are supported for the factory-reset command:

  - **bypass-secure-erase** : Use this option when secure data removal is not required (repartition and reformat storage only).

    preserve-image : This option preserves the running image and autoboots after the completion of erase operations.

After the factory reset process is successfully completed, the switch reboots automatically and powers down. All data specified for secure erase is removed.

What to do next

You can boot the secure erase image to trigger the data wipe.

Secure erase examples

This section comprises examples of secure erase factory reset command and FEX logs.

This is an example output for configuring secure erase factory reset command.

```
FX2-2- switch#
FX2-2- switch# show fex
FEX              FEX              FEX              FEX
Number          Description      State            Model
Serial
------------------------------------------------------------------------
109             FEX0109          Online          N2K-C2348TQ-10GE
FOC1816R0F2
110             FEX0110          Online          N2K-C2348TQ-10G-E
FOC2003R1SQ
```

```
FX2-2-switch# factory-reset fex all
!!!! WARNING:
This command will perform factory-reset of all FEX modules !!!!
The factory reset operation will erase ALL persistent storage on the specified FEX module.
This includes configuration, all log data, and the full contents of flash and SSDs.
Special steps are taken in an effort to render data non-recoverable. Please, proceed with
caution and understanding that this operation cannot be undone and will leave the system
in a fresh-from-factory state.
!!!! WARNING !!!!

Do you want to continue? (y/n) [n] y
Initiating factory-reset for the FEX: 109 --- SUCCESS!!
FEX: 109 is reloading for the reset operation to proceed.
Factory reset may take time...
Please, wait and do not power off the FEX...
Trying to remove the FEX:109 config !!!
Initiating factory-reset for the FEX: 110 --- SUCCESS!!
FEX: 110 is reloading for the reset operation to proceed.
Factory reset may take time...
Please, wait and do not power off the FEX...
Trying to remove the FEX:110 config !!!
Successfully removed FEX:110 config. !!!
```

This is an example of FEX log.

```
FX2-2-switch# 2021
FEX console logs:
================
bgl-ads-4157:138> telnet 10.127.118.15 2007
Trying 10.127.118.15...
Connected to 10.127.118.15.
Escape character is '^]'.

fex-109#
fex-109# [129266.313614] writing reset reason 9, Factory-reset requested by abc
[129266.391801] Restarting system - Factory-reset requested by abc [9]
U-Boot 2011.12 (Jun 25 2014 - 16:28:41) Cisco Systems
CPU0: P1020E, Version: 1.1, (0x80ec0011)
Core: E500, Version: 5.1, (0x80212051)
Clock Configuration:
CPU0:666.667 MHz, CPU1:666.667 MHz,
CCB:333.333 MHz,
```

```
DDR:333.333 MHz (666.667 MT/s data rate) (Asynchronous), LBC:83.333 MHz
L1: D-cache 32 kB enabled
I-cache 32 kB enabled
Board: P1020FEX
[MCPSUMR 0x00000000, RSTRSCR 0x00000000, AUTORSTSR 0x0000c000]
I2C buses: ready
Golden image
U-boot retry count 0
Jump to upgradeable image at 0xefd20040
U-Boot 2011.12 (Jun 25 2014 - 16:19:54) Cisco Systems
CPU0: P1020E, Version: 1.1, (0x80ec0011)
Core: E500, Version: 5.1, (0x80212051)
Clock Configuration:
CPU0:666.667 MHz, CPU1:666.667 MHz,
CCB:333.333 MHz,
DDR:333.333 MHz (666.667 MT/s data rate) (Asynchronous), LBC:83.333 MHz
L1: D-cache 32 kB enabled
I-cache 32 kB enabled
Board: P1020FEX
[MCPSUMR 0x00000000, RSTRSCR 0x00000000, AUTORSTSR 0x0000c000]
I2C buses: ready
Upgradeable image
DRAM: Configuring DDR for 666.667 MT/s data rate
Time-out count = 480
DDR configuration get done
1 GiB (DDR3, 32-bit, CL=6, ECC on)
Memory test from 0x40000 to 0x1fdfffff
Data line test.................. OK
Address line test.............. OK
OK
Flash: 288 MiB
L2: 256 KB enabled
Set dbglevel to its default value (0x1)
PCIe1: Root Complex of mini PCIe SLOT, x1, regs @ 0xffe0a000
PCIe1: Bus 00 - 01
PCIe2: Root Complex of PCIe SLOT, no link, regs @ 0xffe09000
PCIe2: Bus 02 - 02
Net: eTSEC1, eTSEC3
Hit Ctrl-L to stop autoboot: 0
WARN: user forced bootcmd="run sysboot"
.. WARNING: adjusting available memory to 30000000
## Booting kernel from Legacy Image at 01000000 ...
Image Name: Linux-2.6.27.47
Created: 2015-11-20 10:22:39 UTC
Image Type: PowerPC Linux Kernel Image (gzip compressed)
Data Size: 8936305 Bytes = 8.5 MiB
Load Address: 00000000
Entry Point: 00000000
Verifying Checksum ... OK
## Flattened Device Tree blob at 00c00000
Booting using the fdt blob at 0x00c00000
Uncompressing Kernel Image ... OK
Loading Device Tree to 03ffb000, end 03fffe82 ... OK
setup_arch: bootmem
mpc85xx_fex_setup_arch()
arch: exit
[0.436112] Host controller irq 17
[0.477490] pci 0000:00:00.0: ignoring class b20 (doesn't match header type 01)
[0.566841] Assign root port irq 17 for 0000:00:00.0
[2.210329] Enabling all PCI devices
[2.802226] FSL:i2c-mpc - probing i2c controller
modprobe: FATAL: Could not load /lib/modules/2.6.27.47/modules.dep: No such file or directory
[2.975494] FSL:i2c-mpc - probing i2c controller
modprobe: FATAL: Could not load /lib/modules/2.6.27.47/modules.dep: No such file or directory
```

```
modprobe: FATAL: Could not load /lib/modules/2.6.27.47/modules.dep: No such file or directory
modprobe: FATAL: Could not load /lib/modules/2.6.27.47/modules.dep: No such file or directory
modprobe: FATAL: Could not load /lib/modules/2.6.27.47/modules.dep: No such file or directory
modprobe: FATAL: Could not load /lib/modules/2.6.27.47/modules.dep: No such file or directory
modprobe: FATAL: Could not load /lib/modules/2.6.27.47/modules.dep: No such file or directory
modprobe: FATAL: Could not load /lib/modules/2.6.27.47/modules.dep: No such file or directory
[3.889037]
[3.889041] Watchdog init<0>
Mount failed for selinuxfs on /selinux: No such file or directory
INIT: version 2.86 booting
Setting system clock: [ OK ]
Mounting all filesystems: [ OK ]
/sbin/dhclient-script: configuration for eth1 not found. Continuing with defaults.
/etc/sysconfig/network-scripts/network-functions: line 78: eth1: No such file or directory
Mounting system image: [ OK ]
Unpacking system image: [ OK ]
Uncompressing system image: [ OK ]
Loading system image: [ OK ]
net.ipv4.ip_forward = 0
net.ipv4.ip_default_ttl = 64
net.ipv4.ip_no_pmtu_disc = 1
Starting internet superserver: inetd [ OK ]
net.core.rmem_max = 524288
net.core.wmem_max = 524288
net.core.rmem_default = 524288
net.core.wmem_default = 524288
net.core.somaxconn = 1024
net.core.netdev_max_backlog = 1024
modprobe: FATAL: Could not load /lib/modules/2.6.27.47/modules.dep: No such file or directory
[23.255118] Device eth0 configured with sgmii interface
Non issu restart
[24.151321]
[24.151327] base_addr is 26524<0>
Secure erase requested! Please, do not power off module!
Starting the secure erase. !!
This may take time. Please wait !!
>>>> Wiping all storage devices ...
[28.706882] NX-OS starts punching watchdog
grep: Backu: No such file or directory
+++ Starting mtd secure erase for the partition /dev/mtd2 +++
Erasing /dev/mtd2 ...
Erasing 128 Kibyte @ 17e0000 -- 99 % complete.
---> SUCCESS
Writing random data onto /dev/mtd2
Filling /dev/mtd2 using random data ...
Erasing blocks: 192/192 (100%)
Writing data: 24576k/24576k (100%)
Verifying data: 24576k/24576k (100%)
---> SUCCESS
Erasing /dev/mtd2 ...
Erasing 128 Kibyte @ 17e0000 -- 99 % complete.
---> SUCCESS
+++ Skipping cmos secure erase +++
>>>> Done
+++ Skipping nvram secure erase +++
>>>> Done
>>>> Iniatilzing system to factory defaults ...
+++ Starting init-system +++
Initializing /dev/mtd5
/isan/bin/mount_jffs2.sh: line 68: ${LOG_FILE}: ambiguous [ 651.954326] Restarting system.
U-Boot 2011.12 (Jun 25 2014 - 16:28:41) Cisco Systems
CPU0: P1020E, Version: 1.1, (0x80ec0011)
Core: E500, Version: 5.1, (0x80212051)
Clock Configuration:
```

```
CPU0:666.667 MHz, CPU1:666.667 MHz,
CCB:333.333 MHz,
DDR:333.333 MHz (666.667 MT/s data rate) (Asynchronous), LBC:83.333 MHz
L1: D-cache 32 kB enabled
I-cache 32 kB enabled
Board: P1020FEX
[MCPSUMR 0x00000000, RSTRSCR 0x00000000, AUTORSTSR 0x0000c000]
I2C buses: ready
Golden image
U-boot retry count 1
Jump to upgradeable image at 0xefd20040
U-Boot 2011.12 (Jun 25 2014 - 16:19:54) Cisco Systems
CPU0: P1020E, Version: 1.1, (0x80ec0011)
Core: E500, Version: 5.1, (0x80212051)
Clock Configuration:
CPU0:666.667 MHz, CPU1:666.667 MHz,
CCB:333.333 MHz,
DDR:333.333 MHz (666.667 MT/s data rate) (Asynchronous), LBC:83.333 MHz
L1: D-cache 32 kB enabled
I-cache 32 kB enabled
Board: P1020FEX
[MCPSUMR 0x00000000, RSTRSCR 0x00000000, AUTORSTSR 0x0000c000]
I2C buses: ready
Upgradeable image
DRAM: Configuring DDR for 666.667 MT/s data rate
Time-out count = 480
DDR configuration get done
1 GiB (DDR3, 32-bit, CL=6, ECC on)
Memory test from 0x40000 to 0x1fdfffff
Data line test..... OK
Address line test..... OK
OK
Flash: 288 MiB
L2: 256 KB enabled
Set dbglevel to its default value (0x1)
PCIe1: Root Complex of mini PCIe SLOT, x1, regs @ 0xffe0a000
PCIe1: Bus 00 - 01
PCIe2: Root Complex of PCIe SLOT, no link, regs @ 0xffe09000
PCIe2: Bus 02 - 02
Net: eTSEC1, eTSEC3
Hit Ctrl-L to stop autoboot: 0
WARN: user forced bootcmd="run sysboot"
.. WARNING: adjusting available memory to 30000000
## Booting kernel from Legacy Image at 01000000 ...
Image Name: Linux-2.6.27.47
Created: 2015-11-20 10:22:39 UTC
Image Type: PowerPC Linux Kernel Image (gzip compressed)
Data Size: 8936305 Bytes = 8.5 MiB
Load Address: 00000000
Entry Point: 00000000
Verifying Checksum ... OK
## Flattened Device Tree blob at 00c00000
Booting using the fdt blob at 0x00c00000
Uncompressing Kernel Image ... OK
Loading Device Tree to 03ffb000, end 03fffe82 ... OK
setup_arch: bootmem
mpc85xx_fex_setup_arch()
arch: exit
[ 0.436112] Host controller irq 17
[ 0.477490] pci 0000:00:00.0: ignoring class b20 (doesn't match header type 01)
[ 0.566841] Assign root port irq 17 for 0000:00:00.0
[ 2.210556] Enabling all PCI devices
[ 2.804559] FSL:i2c-mpc - probing i2c controller
modprobe: FATAL: Could not load /lib/modules/2.6.27.47/modules.dep: No such file or directory
```

```
[ 2.975502] FSL:i2c-mpc - probing i2c controller
modprobe: FATAL: Could not load /lib/modules/2.6.27.47/modules.dep: No such file or directory
modprobe: FATAL: Could not load /lib/modules/2.6.27.47/modules.dep: No such file or directory
modprobe: FATAL: Could not load /lib/modules/2.6.27.47/modules.dep: No such file or directory
modprobe: FATAL: Could not load /lib/modules/2.6.27.47/modules.dep: No such file or directory
modprobe: FATAL: Could not load /lib/modules/2.6.27.47/modules.dep: No such file or directory
modprobe: FATAL: Could not load /lib/modules/2.6.27.47/modules.dep: No such file or directory
modprobe: FATAL: Could not load /lib/modules/2.6.27.47/modules.dep: No such file or directory
[ 3.889014]
[ 3.889018] Watchdog init<0>
Mount failed for selinuxfs on /selinux: No such file or directory
INIT: version 2.86 booting
Setting system clock: [ OK ]
Mounting all filesystems: [ OK ]
/sbin/dhclient-script: configuration for eth1 not found. Continuing with defaults.
/etc/sysconfig/network-scripts/network-functions: line 78: eth1: No such file or directory
Mounting system image: [ OK ]
Unpacking system image: [ OK ]
Uncompressing system image: [ OK ]
Loading system image: [ OK ]
net.ipv4.ip_forward = 0
net.ipv4.ip_default_ttl = 64
net.ipv4.ip_no_pmtu_disc = 1
Starting internet superserver: inetd [ OK ]
net.core.rmem_max = 524288
net.core.wmem_max = 524288
net.core.rmem_default = 524288
net.core.wmem_default = 524288
net.core.somaxconn = 1024
net.core.netdev_max_backlog = 1024
modprobe: FATAL: Could not load /lib/modules/2.6.27.47/modules.dep: No such file or directory
[ 22.630994] Device eth0 configured with sgmii interface
Non issu restart
[ 23.535827]
[ 23.535832] base_addr is 26524<0>
INIT: Entering runlevel: 3
fex login: Sorry, user root is not allowed to execute '/sbin/sysctl -q -w vm.drop_caches=3'
 as root on fex.
[ 28.090052] NX-OS starts punching watchdog
fex login:
```

This is an example output for configuring secure erase factory reset command on a module:

```
switch# factory-reset [all | module <mod>]
switch# factory-reset [module <3>]
!!!! WARNING !!!!
The factory reset operation will erase ALL persistent storage on the specified module.
This includes configuration, all log data, and the full contents of flash and SSDs.
Special steps are taken to render data non-recoverable. Please, proceed with caution and
understanding that this operation cannot be undone and will leave the system in a
fresh-from-factory state.
!!!! WARNING !!!!
Continue? (y/n) [n] y
A module reload is required for the reset operation to proceed. Please, wait...
…truncated…
Secure erase requested! Please, do not power off module!
>>>> Wiping all storage devices ...
+++ Starting mmc secure erase for /dev/mmcblk0 +++
*** Please, wait - this may take several minutes ***
\
---> SUCCESS
+++ Starting SSD secure erase for /dev/sda +++
*** Please, wait - this may take several minutes ***
\
---> SUCCESS
```

```
+++ Starting cmos secure erase +++
\
---> SUCCESS
>>>> Done
+++ Starting nvram secure erase +++
\
---> SUCCESS
>>>> Done
```

This is an example output log for configuring secure erase factory reset command on a line card.

```
switch# show mod
Mod    Ports        Module-Type                        Model        Status
----   ------     -------------------------------  -------------  ---------
1        32          32x40/100G Ethernet Module    N9K-X9732C-FX    ok
22        0           4-slot Fabric Module         N9K-C9504-FM-E   ok
24        0           4-slot Fabric Module         N9K-C9504-FM-E   ok
26        0           4-slot Fabric Module         N9K-C9504-FM-E   ok
27        0           Supervisor Module            N9K-SUP-B+       active *
28        0           Supervisor Module            N9K-SUP-B+       ha-standby
29        0           System Controller            N9K-SC-          active
30        0           System Controller            N9K-SC-          standby

Mod        Sw             Hw         Slot
----------------------------------------
1         10.2(1.196)     0.1070     LC1
22        10.2(1.196)     1.2        FM2
24        10.2(1.196)     1.2        FM4
26        10.2(1.196)     1.1        FM6
27        10.2(1.196)     1.0        SUP1
28        10.2(1.196)     1.2        SUP2
29        10.2(1.196)     1.4        SC1
30        10.2(1.196)     1.4        SC2

switch#
switch# factory-reset mod 1
!!!! WARNING !!!!
The factory reset operation will erase ALL persistent storage on the specified module.
This includes configuration, all log data, and the full contents of flash and SSDs.
Special steps are taken in an effort to render data non-recoverable.
Please, proceed with
caution and understanding that this operation cannot be undone and will leave the system
in
a fresh-from-factory state.
!!!! WARNING !!!!
Continue? (y/n) [n] y
A module reload is required for the reset operation to proceed. Please, wait...
reloading module 1 ...
.....................................................................
SUCCESS! All persistent storage devices detected on the specified module have been purged.

switch#
switch# show mod
Mod             Ports      Module-Type                    Model             Status
-------      ---------   ----------------------------    --------------    ------------
1                32        32x40/100G Ethernet Module    N9K-X9732C-FX       powered-dn
22                0            4-slot Fabric Module      N9K-C9504-FM-E        ok
24                0            4-slot Fabric Module      N9K-C9504-FM-E        ok
26                0            4-slot Fabric Module      N9K-C9504-FM-E        ok
27                0            Supervisor Module         N9K-SUP-B+           active *
28                0            Supervisor Module         N9K-SUP-B+           ha-standby
29                0            System Controller         N9K-SC-A             active
30                0            System Controller         N9K-SC-A             standby

Mod     Power-Status         Reason
```

```
------  ----------------    ------------
1         powered-dn        Configured Power down

Mod     Sw                 Hw         Slot
22      10.2(1.196)        1.2        FM2
24      10.2(1.196)        1.2        FM4
26      10.2(1.196)        1.1        FM6
27      10.2(1.196)        1.0        SUP1
28      10.2(1.196)        1.2        SUP2
29      10.2(1.196)        1.4        SC1
switch#
```

This is an example output log for configuring secure erase factory reset command on a module.

```
switch# factory-reset mod 26
!!!! WARNING !!!!
The factory reset operation will erase ALL persistent storage on the specified module.
This includes configuration, all log data, and the full contents of flash and SSDs.
Special steps are taken in an effort to render data non-recoverable.
Please, proceed with
caution and understanding that this operation cannot be undone and will leave the system
in
a fresh-from-factory state.
!!!! WARNING !!!!
Continue? (y/n) [n] y
A module reload is required for the reset operation to proceed. Please, wait...
reloading module 26 ...
.............................................................................
..........................
.............................................................................
..........................
.............................................................................
..........................
.............................................................................
..........................
.............................................................................
..........................
.............................................................................
..........................
...................................................................
SUCCESS! All persistent storage devices detected on the specified module have been cleared.
>>>> Please, note - multiple write passes were required to remove data from one or more
devices. <<<<

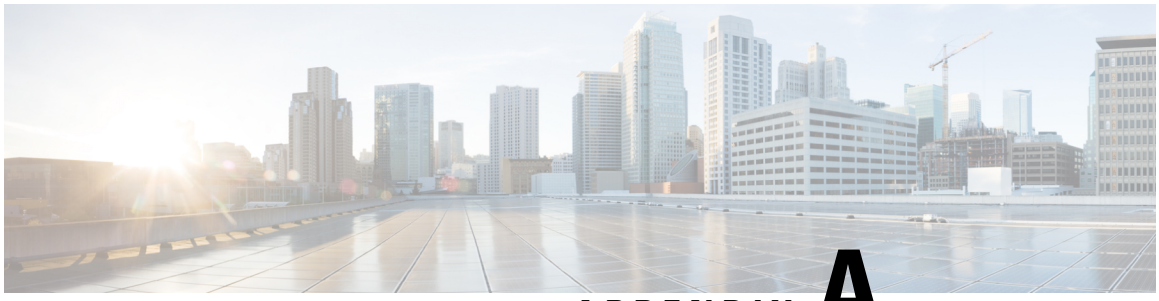
switch# show mod

Mod       Ports      Module-Type                  Model             Status
------  --------    ---------------------------  -------------    ------------
1         32         32x40/100G Ethernet Module  N9K-X9732C-FX     powered-dn
22         0         4-slot Fabric Module        N9K-C9504-FM-E     ok
24         0         4-slot Fabric Module        N9K-C9504-FM-E     ok
26         0         4-slot Fabric Module        N9K-C9504-FM-E    powered-dn

Mod         Power-Status       Reason
------     ---------------    ------------------------
1            powered-dn        Configured Power down
26           powered-dn        Configured Power down

Mod    Sw            Hw           Slot
----  ------       ---------    -----------
22    10.2(1.196)    1.2            FM2
24    10.2(1.196)    1.2            FM4
27    10.2(1.196)    1.0            SUP1
28    10.2(1.196)    1.2            SUP2
```

```
29    10.2(1.196)    1.4            SC1
switch#
```



APPENDIX A

IETF RFCs supported by Cisco NX-OS System Management

This appendix lists the IETF RFCs for system management supported in Cisco NX-OS.

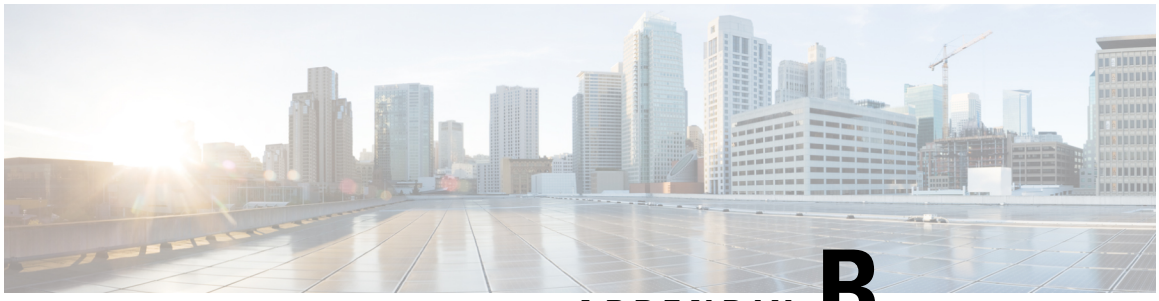
• IETF RFCs supported by Cisco NX-OS System Management , on page 657

IETF RFCs supported by Cisco NX-OS System Management

This appendix provides the IETF RFCs for system management that Cisco NX-OS supports.

RFCs	Title
RFC 2819	*Remote Network Monitoring Management Information Ba*
RFC 3164	*The BSD syslog Protocol*
RFC 3411 and RFC 3418	*An Architecture for Describing Simple Network Managem (SNMP) Management Frameworks*

IETF RFCs supported by Cisco NX-OS System Management



APPENDIX B

Embedded Event Manager System Events and Configuration Examples

- System Policies, on page 659
- Events, on page 661
- Configuration Examples, on page 662

System Policies

The following table lists the EEM system policies, the events they monitor, and the corresponding automated actions taken by the system.

Table 46: EEM System Policies and Actions

Event	Description / Action
__BootupPortLoopback	Do CallHome, Error-disable affected ports, log error testing on affected ports after 1 consecutive failure of GOLD "BootupPortLoopback" test.
__PortLoopback	Do CallHome, log error in Syslog/OBFL/Exception Log, and disable further HM testing on affected ports after 10 consecutive failures of GOLD "PortLoopback" test.
__RewriteEngineLoopback	Do CallHome, log error in Syslog/OBFL/Exception Log, and disable further HM testing on affected ports after 10 consecutive failures of GOLD "RewriteEngine" test.
__asicmem	Do CallHome and log error when GOLD "AsicMemory" test fails. As the issue causing the test failure may be transient, attempt recovery reload through kernel panic. **Note** To avoid a kernel panic when the test fails, you can override the EEM system policy.
__asic_register_check	Do CallHome, log error, and disable further HM testing for that ASIC device/instance after 20 consecutive failures of GOLD "ASICRegisterCheck" test.

Event	Description / Action
__compact_flash	Do CallHome, log error, and disable further HM testing after 20 consecutive failures of GOLD "CompactFlash" test.
__crypto_device	Do CallHome and log error when GOLD "CryptoDevice" test fails.
__eobc_port_loopback	Do CallHome and log error when GOLD "EOBCPortLoopback" test fails.
__ethpm_debug_1	Action: none
__ethpm_debug_2	Action: none
__ethpm_debug_3	Action: none
__ethpm_debug_4	Action: none
__ethpm_link_flap	More than 30 link flaps in a 420-second interval. Action: Error. Disable the port.
__external_compact_flash	Do CallHome, log error, and disable further HM testing after 20 consecutive failures of GOLD "ExternalCompactFlash" test.
__fpgareg	Do CallHome, log error, disable further HM testing after 20 consecutive failures of GOLD "FpgaRegTest" test. As the issue causing the test failure may be transient, attempt recovery reload through kernel panic. **Note** To avoid a kernel panic when the test fails, you can override the EEM system policy.
__L2ACLRedirect	Do CallHome, log error, disable further HM testing after 10 consecutive failures of L2ACLRedirect test. As the issue causing the test failure may be transient, attempt recovery reload through kernel panic. **Note** To avoid a kernel panic when the test fails, you can override the EEM system policy.
__lcm_module_failure	Power cycle two times and then power down.
__management_port_loopback	Do CallHome and log error when GOLD "ManagementPortLoopback" test fails.
__nvram	Do CallHome, log error, and disable further HM testing after 20 consecutive failures of GOLD "NVRAM" test.
__pfm_fanabsent_all_systemfan	Shuts down if both fan trays (f1 and f2) are absent for 2 minutes.
__pfm_fanbad_all_systemfan	Syslog when fan goes bad.
__pfm_fanbad_any_singlefan	Syslog when fan goes bad.
__pfm_power_over_budget	Syslog warning for insufficient power overbudget.
__pfm_tempev_major	TempSensor Major Threshold. Action: Shutdown.

Event	Description / Action
__pfm_tempev_minor	TempSensor Minor Threshold. Action: Syslog.
__primary_bootrom	Do CallHome, log error, and disable further HM testing after 20 consecutive failures of GOLD "PrimaryBootROM" test.
__pwr_mgmt_bus	Do CallHome, log error, and disable further HM testing for the module or spine-card after 20 consecutive failures of GOLD "PwrMgmtBus" test.
__real_time_clock	Do CallHome, log error, and disable further HM testing after 20 consecutive failures of GOLD "RealTimeClock" test.
__secondary_bootrom	Do CallHome, log error, and disable further HM testing after 20 consecutive failures of GOLD "SecondaryBootROM" test.
__spine_control_bus	Do CallHome, log error, and disable further HM testing for that module or spine-card after 20 consecutive failures of GOLD "SpineControlBus" test.
__standby_fabric_loopback	Do CallHome, log error, and disable further HM testing after 10 consecutive failures.
__status_bus	Do CallHome, log error, and disable further HM testing after 5 consecutive failures of GOLD "StatusBus" test.
__system_mgmt_bus	Do Call Home, log error, and disable further HM testing for that fan or power supply after 20 consecutive failures of GOLD "SystemMgmtBus" test.
__usb	Do Call Home and log error when GOLD "USB" test fails.

Events

The following table describes the EEM events you can use on the device.

Table 47: EEM Events Lookup Table

EEM Event	Description
application	Publishes an application-specific event.
cli	CLI command is entered that matches a pattern with a wildcard.
counter	EEM counter reaches a specified value or range.
fanabsent	System fan tray is absent.
fanbad	System fan generates a fault.
fib	Monitors routes or TCAM usage in the unicast FIB.
gold	GOLD test failure condition is hit.

EEM Event	Description
interface	Interface counter exceeds a threshold.
memory	Available system memory exceeds a threshold.
module	Specified module enters the selected status.
module-failure	Module failure is generated.
none	Runs the policy event without any events specified.
oir	Online insertion or removal occurs.
policy-default	Default parameters and thresholds are used for the events in the system policy you override.
poweroverbudget	Platform software detects a power budget condition.
snmp	SNMP object ID (OID) state changes.
storm-control	Platform software detects an Ethernet packet storm condition.
syslog	Monitors syslog messages and invokes the policy based on the search string in the policy.
sysmgr	System manager generates an event.
temperature	Temperature level in the system exceeds a threshold.
timer	Specified time is reached.
track	Tracked object changes state.

Configuration Examples

Command Line Interface Events

- Monitor Interface Shutdown
- Monitor Module Powerdown
- Add Trigger to Initiate a Rollback

Monitor Interface Shutdown

This example shows how to monitor an interface shutdown:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# event manager applet monitorShutdown
switch(config-applet)#
```

```
switch(config-applet)# description "Monitors interface shutdown."
switch(config-applet)# event cli match "conf t; interface *; shutdown"
switch(config-applet)# action 1.0 cli show interface e 3/1
switch(config)# copy running-config startup-config
```

Note Outputs of **show** commands entered as part of EEM policy are archived in the logflash as text files with the "eem_archive_" prefix. To view the archived output, use the **show file logflash:eem_archive_***n* command.

Monitor Module Powerdown

This example shows how to monitor a module powerdown:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# event manager applet monitorPoweroff
switch(config-applet)#
switch(config-applet)# description "Monitors module power down."
switch(config-applet)# event cli match "conf t; poweroff *"
switch(config-applet)# action 1.0 cli show module
switch(config)# copy running-config startup-config
```

Add Trigger to Initiate a Rollback

This example shows how to add a trigger to initiate a rollback:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)#
switch(config)# event manager applet rollbackTrigger
switch(config-applet)#
switch(config-applet)# description "Rollback trigger."
switch(config-applet)# event cli match "rollback *"
switch(config-applet)# action 1.0 cli copy running-config bootflash:last_config
switch(config)# copy running-config startup-config
```

Override (Disable) Major Thresholds

- Prevent Shutdown When Reaching a Major Threshold
- Disable One Bad Sensor
- Disable Multiple Bad Sensors
- Override (Disable) an Entire Module
- Override (Disable) Multiple Modules and Sensors
- Enable One Sensor While Disabling All Remaining Sensors of All Modules
- Enable Multiple Sensors While Disabling All Remaining Sensors of All Modules
- Enable All Sensors of One Module While Disabling All Sensors of the Remaining Modules
- Enable Combination of Sensors on Modules While Disabling All Sensors of the Remaining Modules

Prevent Shutdown When Reaching a Major Threshold

This example shows how to prevent a shutdown caused by reaching a major threshold:

```
switch# configure terminal
switch(config)# event manager applet myappletname override __pfm_tempev_major
switch(config)# end
```

This example shows how to revert to the default configuration:

```
switch# configure terminal
switch(config)# no event manager applet myappletname override __pfm_tempev_major
switch(config)# end
```

Disable One Bad Sensor

This example shows how to disable only sensor 3 on module 2 when sensor 3 is malfunctioning (all other sensors are unaffected):

```
switch# configure terminal
switch(config)# event manager applet myappletname override __pfm_tempev_major
switch(config-applet)# event temperature module 2 sensor 3 threshold major
switch(config-applet)# end
```

This example shows how to revert to the default configuration:

```
switch# configure terminal
switch(config)# no event manager applet myappletname override __pfm_tempev_major
switch(config)# end
```

Disable Multiple Bad Sensors

This example shows how to disable sensors 5, 6, and 7 on module 2 when these sensors are malfunctioning (all other sensors are unaffected):

```
switch# configure terminal
switch(config)# event manager applet myappletname override __pfm_tempev_major
switch(config-applet)# event temperature module 2 sensor 5 threshold major
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myappletname override __pfm_tempev_major
switch(config-applet)# event temperature module 2 sensor 6 threshold major
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myappletname override __pfm_tempev_major
switch(config-applet)# event temperature module 2 sensor 7 threshold major
switch(config-applet)# end
```

This example shows how to revert to the default configuration:

```
switch# configure terminal
switch(config)# no event manager applet myappletname override __pfm_tempev_major
switch(config)# end
```

Override (Disable) an Entire Module

This example shows how to disable module 2 when it is malfunctioning:

```
switch# configure terminal
switch(config)# event manager applet myappletname override __pfm_tempev_major
switch(config-applet)# event temperature module 2 threshold major
switch(config-applet)# end
```

This example shows how to revert to the default configuration:

```
switch# configure terminal
switch(config)# no event manager applet myappletname override __pfm_tempev_major
switch(config)# end
```

Override (Disable) Multiple Modules and Sensors

This example shows how to disable sensors 3, 4, and 7 on module 2 and all sensors on module 3 when they are malfunctioning:

```
switch# configure terminal
switch(config)# event manager applet myappletname override __pfm_tempev_major
switch(config-applet)# event temperature module 2 sensor 3 threshold major
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myappletname override __pfm_tempev_major
switch(config-applet)# event temperature module 2 sensor 4 threshold major
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myappletname override __pfm_tempev_major
switch(config-applet)# event temperature module 2 sensor 7 threshold major
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myappletname override __pfm_tempev_major
switch(config-applet)# event temperature module 3 threshold major
switch(config-applet)# end
```

This example shows how to revert to the default configuration:

```
switch# configure terminal
switch(config)# no event manager applet myappletname override __pfm_tempev_major
switch(config)# end
```

Enable One Sensor While Disabling All Remaining Sensors of All Modules

This example shows how to disable all sensors on all modules except sensor 4 on module 9:

```
switch# configure terminal
switch(config)# event manager applet myapplet1 override __pfm_tempev_major
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet2 override __pfm_tempev_major
switch(config-applet)# event temperature module 9 sensor 4 threshold major
switch(config-applet)# action 2 policy-default
switch(config-applet)# end
```

Enable Multiple Sensors While Disabling All Remaining Sensors of All Modules

This example shows how to disable all sensors on all modules except sensors 4, 6, and 7 on module 9:

```
switch# configure terminal
switch(config)# event manager applet myapplet1 override __pfm_tempev_major
```

```
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet2 override __pfm_tempev_major
switch(config-applet)# event temperature module 9 sensor 4 threshold major
switch(config-applet)# action 2 policy-default
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet3 override __pfm_tempev_major
switch(config-applet)# event temperature module 9 sensor 6 threshold major
switch(config-applet)# action 3 policy-default
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet4 override __pfm_tempev_major
switch(config-applet)# event temperature module 9 sensor 7 threshold major
switch(config-applet)# action 4 policy-default
switch(config-applet)# end
```

Enable All Sensors of One Module While Disabling All Sensors of the Remaining Modules

This example shows how to disable all sensors on all modules except all sensors on module 9:

```
switch# configure terminal
switch(config)# event manager applet myapplet1 override __pfm_tempev_major
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet2 override __pfm_tempev_major
switch(config-applet)# event temperature module 9 threshold major
switch(config-applet)# action 2 policy-default
switch(config-applet)# end
```

Enable Combination of Sensors on Modules While Disabling All Sensors of the Remaining Modules

This example shows how to disable all sensors on all modules except sensors 3, 4, and 7 on module 2 and all sensors on module 3:

```
switch# configure terminal
switch(config)# event manager applet myapplet1 override __pfm_tempev_major
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet2 override __pfm_tempev_major
switch(config-applet)# event temperature module 2 sensor 3 threshold major
switch(config-applet)# action 2 policy-default
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet3 override __pfm_tempev_major
switch(config-applet)# event temperature module 2 sensor 4 threshold major
switch(config-applet)# action 3 policy-default
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet4 override __pfm_tempev_major
switch(config-applet)# event temperature module 2 sensor 7 threshold major
switch(config-applet)# action 4 policy-default
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet5 override __pfm_tempev_major
switch(config-applet)# event temperature module 3 threshold major
switch(config-applet)# action 5 policy-default
switch(config-applet)# end
```

Override (Disable) Shutdown for Fan Tray Removal

- Override (Disable) Shutdown for Removal of a Specified Fan Tray
- Override (Disable) Shutdown for Removal of One or More Fan Trays
- Override (Disable) Shutdown for Removal of All Fan Trays Except One from a Set of Fan Trays
- Override (Disable) Shutdown for Removal of Fan Trays Except for a Specified Set of Fan Trays
- Override (Disable) Shutdown for Removal of All Fan Trays Except One
- Override (Disable) Shutdown for Removal of Multiple Specified Fan Trays

Override (Disable) Shutdown for Removal of a Specified Fan Tray

This example shows how to disable a shutdown so that you can remove a specified fan tray (fan tray 3):

```
switch# configure terminal
switch(config)# event manager applet myappletname override __pfm_fanabsent_any_singlefan
switch(config-applet)# event fanabsent fan 3 time 60
switch(config-applet)# end
```

This example shows how to revert to the default configuration:

```
switch# configure terminal
switch(config) no event manager applet myappletname override __pfm_fanabsent_any_singlefan
switch(config)# end
```

Override (Disable) Shutdown for Removal of One or More Fan Trays

This example shows how to disable a shutdown so that you can remove one or more (or all) fan trays:

```
switch# configure terminal
switch(config)# event manager applet myappletname override __pfm_fanabsent_any_singlefan
switch(config-applet)# end
```

This example shows how to revert to the default configuration:

```
switch# configure terminal
switch(config)# no event manager applet myappletname override __pfm_fanabsent_any_singlefan
switch(config-applet)# end
```

Override (Disable) Shutdown for Removal of All Fan Trays Except One from a Set of Fan Trays

This example shows how to disable a shutdown so that you can remove all fan trays except one from a set of fan trays (fan trays 2, 3, or 4):

```
switch# configure terminal
switch(config)# event manager applet myapplet1 override __pfm_fanabsent_any_singlefan
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet2 override __pfm_fanabsent_any_singlefan
switch(config-applet)# event fanabsent fan 2 time 60
switch(config-applet)# action 2 policy-default
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet3 override __pfm_fanabsent_any_singlefan
switch(config-applet)# event fanabsent fan 3 time 60
```

```
switch(config-applet)# action 3 policy-default
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet4 override __pfm_fanabsent_any_singlefan
switch(config-applet)# event fanabsent fan 4 time 60
switch(config-applet)# action 4 policy-default
switch(config-applet)# end
```

Override (Disable) Shutdown for Removal of Fan Trays Except for a Specified Set of Fan Trays

This example shows how to disable a shutdown so that you can remove fans except for a specified set of fan trays (fan trays 2, 3, and 4):

```
switch# configure terminal
switch(config)# event manager applet myapplet1 override __pfm_fanabsent_any_singlefan
switch(config-applet)# end
switch(config)# event manager applet myapplet2 override __pfm_fanabsent_any_singlefan
switch(config-applet)# event fanabsent fan 2,3,4 time 60
switch(config-applet)# action 2 policy-default
switch(config-applet)# end
```

Override (Disable) Shutdown for Removal of All Fan Trays Except One

This example shows how to disable a shutdown so that you can remove all fan trays except one (fan tray 2):

```
switch# configure terminal
switch(config)# event manager applet myapplet1 override __pfm_fanabsent_any_singlefan
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet2 override __pfm_fanabsent_any_singlefan
switch(config-applet)# event fanabsent fan 2 time 60
switch(config-applet)# action 2 policy-default
switch(config-applet)# end
```

Override (Disable) Shutdown for Removal of Multiple Specified Fan Trays

This example shows how to disable a shutdown so that you can remove multiple specified fan trays (fan trays 2, 3, and 4):

```
switch# configure terminal
switch(config)# event manager applet myapplet1 override __pfm_fanabsent_any_singlefan
switch(config-applet)# event fanabsent fan 2 time 60
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet2 override __pfm_fanabsent_any_singlefan
switch(config-applet)# event fanabsent fan 3 time 60
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet myapplet3 override __pfm_fanabsent_any_singlefan
switch(config-applet)# event fanabsent fan 4 time 60
switch(config-applet)# end
```

This example shows how to revert to the default configuration:

```
switch# configure terminal
switch(config)# no event manager applet myappletname override __pfm_fanabsent_any_singlefan
switch(config)# end
```

Create Supplemental Policy

- Create Supplemental Policy for the Fan Tray Absent Event
- Create Supplemental Policy for the Temperature Threshold Event

Create Supplemental Policy for the Fan Tray Absent Event

This example shows how to create a supplemental policy using the **event fanabsent** command:

[**no**] **event fanabsent** [**fan** *fan-tray-number*] **time** *time-interval*

In addition to the default policy, this example shows how to execute the policy myappletname and action 3 if fan tray 1 is absent for 60 seconds:

```
switch# configure terminal
switch(config)# event manager applet myappletname
switch(config-applet)# event fanabsent fan 1 time 60
switch(config-applet)# action 3 cli "show env fan"
switch(config-applet)# end
```

Create Supplemental Policy for the Temperature Threshold Event

This example shows how to create a supplemental policy using the **event temperature** command:

[**no**] **event temperature** [**mod** *module-number*] [**sensor** *sensor-number*] **threshold** {**major** | **minor** | **any**}

In addition to the default policy, this example shows how to execute the policy myappletname and action 1 if the temperature crosses the minor threshold on sensor 3 of module 2:

```
switch# configure terminal
switch(config)# event manager applet myappletname
switch(config-applet)# event temperature module 2 sensor 3 threshold minor
switch(config-applet)# action 1 cli "show environ temperature"
switch(config-applet)# end
```

Power Over-Budget Policy

The power over-budget policy gets triggered when the available power capacity drops below zero and the device is no longer able to keep the previously powered-up modules in the powered-up state. The default action is to print a syslog to notify the user of the occurrence of power over budget.

You can enable an additional action to power down modules until the available power recovers from the red (negative) zone.

- Shut Down Modules
- Shut Down Specified List of Modules

Shut Down Modules

If you do not specify any modules, the power over-budget shutdown starts from slot 1 and shuts down modules until the power recovers from the red (negative) zone. Empty slots and slots that contain a supervisor, standby supervisor, spine, or crossbar are skipped.

This example shows how to shut down modules starting from module 1 when the available power drops below zero:

```
switch# configure terminal
switch(config)# event manager applet <myappletname4a> override __pfm_power_over_budget
switch(config-applet)# event poweroverbudget
switch(config-applet)# action 4 overbudgetshut
switch(config-applet)# end
```

Shut Down Specified List of Modules

You can specify a list of modules that the power over-budget action uses to shut down modules until the power recovers from the red (negative) zone. Empty slots and slots that contain a supervisor, standby supervisor, spine, or crossbar are skipped.

This example shows how to shut down modules from a specified list of modules (1, 2, 7, 8) when the available power drops below zero:

```
switch# configure terminal
switch(config)# event manager applet <myappletname4b> override __pfm_power_over_budget
switch(config-applet)# event poweroverbudget
switch(config-applet)# action 5 overbudgetshut module 1,2,7,8
switch(config-applet)# end
```

Select Modules to Shut Down

- Use Parameter Substitution to Select Nonoverridden Modules to Shut Down
- Use the Policy Default to Select Nonoverridden Modules to Shut Down

Use Parameter Substitution to Select Nonoverridden Modules to Shut Down

This example shows how to use parameter substitution to select the nonoverridden modules to shut down when a major threshold is exceeded:

```
switch# configure terminal
switch(config)# event manager applet my5b1 override __pfm_tempev_major
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet my5b2 override __pfm_tempev_major
switch(config-applet)# event temperature module 1-3 sensor 8 threshold major
switch(config-applet)# action 6 forceshut module my_module_list reset "temperature-sensor
policy trigger"
switch(config-applet)# end
```

To create event manager parameters, use the **event manager environment** command. To display the values of event manager parameters, use the **show event manager environment all** command.

Use the Policy Default to Select Nonoverridden Modules to Shut Down

This example shows how to use the policy default to select the nonoverridden modules to shut down when a major threshold is exceeded:

```
switch# configure terminal
switch(config)# event manager applet my5a1 override __pfm_tempev_major
switch(config-applet)# end
switch# configure terminal
switch(config)# event manager applet my5a2 override __pfm_tempev_major
switch(config-applet)# event temperature module 1-3 sensor 4 threshold major
switch(config-applet)# action 5 policy-default
switch(config-applet)# end
```

Online Insertion Removal Event

The Online Insertion Removal (OIR) event is a system event that is triggered when a device is inserted or removed while the system is powered on.

- The **event oir** command is used to configure OIR events.
- The *device-type* can be **fan**, **module**, or **powersupply**.
- The *event-type* can be **insert**, **remove**, or **anyoir** (insert or remove).
- The optional *device-number* specifies a single device; if omitted, all devices are selected.

This example shows how to configure the OIR event using the **event oir** command:

event oir *device-type event-type* [*device-number*]

This example shows how to configure the insert event:

```
switch# configure terminal
 switch(config)# event manager applet myoir
 switch(config-applet)# event oir module insert
 switch(config-applet)# action 1 syslog priority critical msg "OIR insert event: A Module
is inserted"
```

This example shows how to configure the remove event:

```
switch# configure terminal
 switch(config)# event manager applet myoir
 switch(config-applet)# event oir module remove
 switch(config-applet)# action 1 syslog priority critical msg "OIR remove event: A Module
is removed"
```

Generate User Syslog

This example shows how to generate a user syslog using the **action syslog** command:

```
 switch# configure terminal
 switch(config)#event manager applet myoir
 witch(config-applet)# event oir module remove
       switch(config-applet)# action 1 syslog priority critical msg "Module is removed"
```

When this event is triggered, the system generates a syslog as follows:

```
switch(config)# 2013 May 20 00:08:27 p1b-57 %$ VDC-1 %$ %EEM_ACTION-2-CRIT: "Module is
removed"
```

Monitor Syslog Messages

```
 switch(config)# event manager applet a1
 switch(config-applet)# event syslog occurs 6 period 4294967 pattern "authentication failed"
```

When this event is triggered, the action defined in the policy is executed.

SNMP Notification

- Poll SNMP OID to Generate an EEM Event
- Send SNMP Notification in Response to an Event in the Event Policy

Poll SNMP OID to Generate an EEM Event

The SNMP object ID (OID) CISCO-SYSTEM-EXT-MIB::cseSysCPUUtilization is used for querying the CPU utilization of the switch:

```
cseSysCPUUtilization OBJECT-TYPE
SYNTAX Gauge32 (0..100 )
UNITS "%"
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The average utilization of CPU on the active supervisor."
::= { ciscoSysInfoGroup 1 }
```

This example shows the use of an SNMP OID that is polled at an interval of 10 seconds and has a threshold value of 95 percent:

```
switch# configure terminal
switch(config)# event manager applet test_policy
switch(config-applet)# event snmp oid 1.3.6.1.4.1.9.9.305.1.1.1.0 get-type exact entry-op
gt entry-val 95 exit-op lt exit-val 90 poll-interval 10
```

Send SNMP Notification in Response to an Event in the Event Policy

You can use this type of configuration to cause a critical event trigger to generate an SNMP notification.

This example shows how to send an SNMP notification for an event from the Event Manager applet configuration mode:

```
switch(config-applet)# action 1.1 snmp-trap intdata1 100 intdata2 300 strdata "CPU Hogging
 at switch1"
switch(config-applet)# action 1.1 snmp-trap intdata1 100 intdata2 300 strdata "Port Failure
 eth9/1"
```

This configuration triggers an SNMP notification (trap) from the switch to SNMP hosts. The SNMP payload carries the values of user-defined fields intdata1, intdata2, and strdata.

Port Tracking

This example shows how to configure the state of one port to match the state of another port (port tracking).

To configure the port tracking of Ethernet interface 3/23 by Ethernet interface 1/2, follow these steps:

Procedure

Step 1 Create an object to track the status of Ethernet interface 3/23.

Example:

```
switch# configure terminal
switch(config)# track 1 interface ethernet 3/23
switch(config-track)# end
```

Step 2 Configure an EEM event to shut Ethernet interface 1/2 when the tracking object shuts down.

Example:

```
switch(config)# event manager applet track_3_23_down
switch(config-applet)# event track 1 state down
switch(config-applet)# action 1 syslog msg EEM applet track_3_23_down shutting down port eth1/2 due
to eth3/23 being down
switch(config-applet)# action 2 cli conf term
switch(config-applet)# action 3 cli interface ethernet 1/2
switch(config-applet)# action 4 cli shut
switch(config-applet)# end
```

Step 3 Configure an EEM event to bring up Ethernet interface 1/2 when Ethernet interface 3/23 comes up.

Example:

```
switch# configure terminal
switch(config)# event manager applet track_3_23_up
switch(config-applet)# event track 1 state up
switch(config-applet)# action 1 syslog msg EEM applet track_3_23_down bringing up port eth1/2 due
to eth3/23 being up
switch(config-applet)# action 2 cli conf term
switch(config-applet)# action 3 cli interface ethernet 1/2
switch(config-applet)# action 4 cli no shut
switch(config-applet)# end
```

Register an EEM Policy with the EEM

This example shows how to register an EEM policy with the EEM:

Basic Switch Configuration



Note The severity keyword is deprecated and only the following patterns are allowed:

[0-9 a-zA-Z][0-9 a-zA-Z]*[-_ ,:/0-9a-zA-Z]*

```
event manager applet vpc_check_peer_at_startup
event track 101 state up
action 1.0 cli copy bootflash:eem/user_script_policies/load_schedules running-config
feature scheduler
!!## 2 x dummy loopbacks are required ##!!
interface loopback 101
interface loopback 102
track 1 list boolean or
object 13
```

```
object 12
object 102
track 2 list boolean and
object 13
object 12
track 12 interface Ethernet 2/24 line-protocol
track 13 interface port-channel 3000 line-protocol
track 101 interface loopback 101 line-protocol
track 102 interface loopback 102 line-protocol
```



Note In this example, port channel 3000 is the vPC peer link, and Ethernet 2/24 is the vPC keepalive link.

You need to copy the following files to the bootflash:

- A directory called: /eem/user_script_policies needs to be created on the supervisor bootflash.
- These five files need to be created and loaded into the above directory:
  - load_schedules
  - remove_vpc_if_peer_failed
  - clean_up
  - unload_schedules
  - restore_vpc

Configuration for the load_schedules File

```
feature scheduler
configure terminal
scheduler job name vpc_check
configure terminal
event manager policy remove_vpc_if_peer_failed
end
configure terminal
scheduler job name clean_up
configure terminal
event manager policy clean_up
end
configure terminal
scheduler job name trigger
configure terminal
interface loopback 102
shutdown
no shutdown
end
configure terminal
scheduler schedule name load_vpc_check
time start +00:00:04
job name vpc_check
scheduler schedule name trigger_vpc_check
time start +00:00:05
job name trigger
scheduler schedule name load_clean_up
time start +00:00:08
job name clean_up
scheduler schedule name trigger_clean_up
```

```
time start +00:00:10
job name trigger
```

Configuration for the remove_vpc_if_peer_failed File

```
event manager applet remove_vpc_if_peer_failed
event track 1 state down
action 1.0 cli show run vpc > bootflash://sup-active/eem/user_script_policies/vpc_saved.cfg
action 2.0 cli show run vpc > bootflash://sup-standby/eem/user_script_policies/vpc_saved.cfg
action 3.0 cli configure terminal
action 4.0 cli no feature vpc
action 5.0 syslog msg severity alert "##### WARNING!!!! PEER SWITCH FAILED TO COME ONLINE.
 VPC CONFIG REMOVED #####"
action 6.0 cli event manager policy restore_vpc
action 7.0 cli copy bootflash:eem/user_script_policies/unload_schedules running-config
action 8.0 cli no event manager applet remove_vpc_if_peer_failed
action 9.0 cli end
```

Configuration for the clean_up File

```
event manager applet clean_up
event track 102 state up
action 1.0 cli configure terminal
action 2.0 cli no event manager applet remove_vpc_if_peer_failed
action 3.0 cli copy bootflash:eem/user_script_policies/unload_schedules running
action 4.0 cli no event manager applet clean_up
action 5.0 end
```

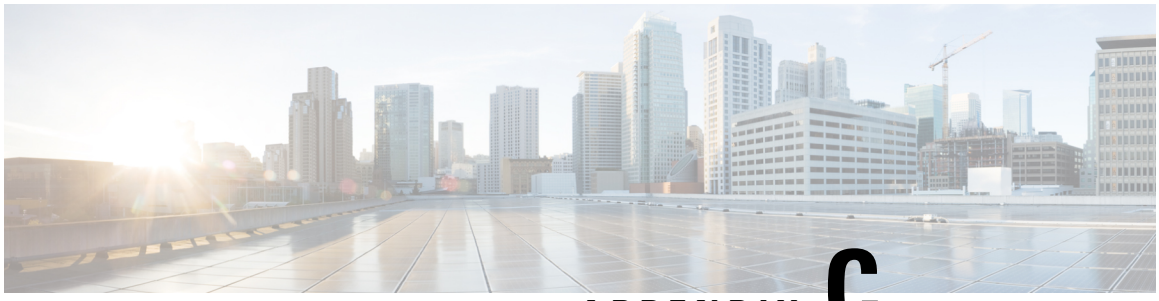
Configuration for the unload_schedules File

```
no scheduler schedule name load_vpc_check
no scheduler schedule name trigger_vpc_check
no scheduler schedule name load_clean_up
no scheduler schedule name trigger_clean_up
no scheduler job name vpc_check
no scheduler job name trigger
no scheduler job name clean_up
```

Configuration for the restore_vpc File

```
event manager applet restore_vpc
event track 2 state up
action 1.0 cli copy bootflash:eem/user_script_policies/vpc_saved.cfg running-config
action 1.0 syslog priority alerts msg VPC PEER DETECTED. VPC CONFIG RESTORED
action 3.0 cli configure terminal
action 4.0 cli copy bootflash:eem/user_script_policies/unload_schedules running-config
action 5.0 cli no event manager applet restore_vpc
action 6.0 cli end
```

Embedded Event Manager System Events and Configuration Examples



APPENDIX C

System Management Configuration Limits in Cisco NX-OS

• System management configuration limits in Cisco NX-OS , on page 677

System management configuration limits in Cisco NX-OS

Each feature supported by Cisco NX-OS has a defined maximum configuration limit. However, a few features have configurations that support limits less than the maximum limits based on a few factors.

The configuration limits are documented in *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide* on Cisco.com.

System Management Configuration Limits in Cisco NX-OS



INDEX

A

abort 32
action 316, 321–322, 324
alert-group {Configuration | Diagnostic | EEM | Environmental | Inventory | License | Supervisor-Hardware | Syslog-group-port | System | Test} user-def-cmd 200

C

callhome 194, 196–197, 199–201, 203–208, 210–211
callhome send 211
callhome send configuration 211
callhome send diagnostic 211
callhome test 211
cdp advertise {v1 | v2} 154
cdp enable 152–153
cdp format device-id {mac-address | serial-number | system-name} 154
cdp holdtime 154
cdp timer 154
cfs ipv4 distribute 27
checkpoint 629
clear cdp counters 152
clear cdp table 152
clear checkpoint database 628
clear counters interface all 501
clear counters mpls strip 526
clear hardware rate-limiter sflow 501
clear lldp counters 431
clear logging logfile 176
clear logging nvram 177
clear logging onboard 365
clear mpls strip label dynamic 526
clear ntp session 132
clear ntp statistics 132
clear sflow statistics 501
Clock protocol gnss 85
collect 458
collect counter 460
collect ip version 460
collect timestamp sys-uptime 460
collect transport tcp flags 460
commit 28, 30–31, 195–196, 198–200, 202–206
config sync 27, 29, 31, 33
configure maintenance profile maintenance-mode 569
configure maintenance profile normal-mode 571
configure session 227
contract-id 195
copy ftp 590
copy tftp 589
customer-id 195

D

description 315, 324, 458, 461–462
destination 461
destination interface 390
destination-profile 196–197, 199, 208
destination-profile {CiscoTAC-1 | full-txt-destination | short-txt-destination} alert-group 199
destination-profile {CiscoTAC-1 | full-txt-destination | short-txt-destination} email-addr 197
destination-profile {CiscoTAC-1 | full-txt-destination | short-txt-destination} http 198
destination-profile {CiscoTAC-1 | full-txt-destination | short-txt-destination} message-level 198
destination-profile {CiscoTAC-1 | full-txt-destination | short-txt-destination} message-size 198
destination-profile {CiscoTAC-1 | full-txt-destination | short-txt-destination} transport-method {email | http} 198
destination-profile name alert-group all 209
destination-profile name email-address email-address 209
diagnostic bootup level {complete | minimal | bypass} 301
diagnostic clear result module 304
diagnostic monitor interval module 302
diagnostic monitor module 302
diagnostic ondemand action-on-failure {continue failure-count | stop} 303
diagnostic ondemand iteration 303
diagnostic start module 303
diagnostic stop module 303
diagnostic test simulation module 304
dir 588
dir bootflash: 591
dscp 461

E

email-contact 194, 208
enable 207, 210

event 315, 324
event application 316
event cli 316
event counter 317
event fanabsent 317
event fanbad 317
event fib adjacency extra 317
event fib resource tcam usage 317
event fib route {extra | inconsistent | missing} 317
event gold module 317
event manager applet 315, 324, 326
event manager environment 314
event manager policy 323
event memory {critical | minor | severe} 317
event module 318
event module-failure 318
event none 318
event oir 318
event policy-default count 318
event poweroverbudget 319
event snmp 319
event storm-control 319
event syslog 319
event syslog {occurs | period | pattern | priority} 326
event syslog tag {occurs | period | pattern | priority} 326
event sysmgr memory 319
event sysmgr switchover count 319
event temperature 320
event timer 320
event track 320
exporter 463

F

feature lldp 429
feature netflow 457
feature ntp 130
feature ptp 80
flow exporter 461
flow monitor 462
flow record 458, 465, 467, 469

G

guestshell sync 606

H

hardware multicast global-tx-span 391

I

import 31
import interface 31
import running-config 31
install activate 592
install add bootflash 591
install add ftp 591
install add tftp 591
install add usb1 591
install add usb2 591
install remove 594
ip access-group 228
ip access-list 227, 516
ip flow monitor 464–465, 467, 469
ip port access-group 519, 523
ipv6 flow monitor 465, 467
isolate 566

J

job name 238

L

layer2-switched flow monitor 466
lldp chassis-id switch 430
lldp dcbx version 434
lldp holdtime 439
lldp receive 432, 439
lldp reinit 440
lldp timer 440
lldp tlv-select 440
lldp transmit 432, 438
logging console 160
logging event {link-status | trunk-status} {enable | default} 163
logging level 165–166
logging logfile 162
logging message interface type ethernet description 161
logging module 164
logging monitor 160
logging origin-id 161
logging server 167, 169
logging source-interface loopback 168
logging timestamp {microseconds | milliseconds | seconds} 166

M

mac access-list 516
mac packet-classify 466
mac port access-group 519, 523
match 458
match datalink 459, 465
match ip 459
match ipv4 459
match ipv6 459
match transport 460
monitor session 396
monitor session all shut 396
monitor session in erspan source 420

mpls strip 522
mpls strip label 524
mpls strip label-age 525
mtu 390

N

NetFlow 464
    bridged on VLAN 464
no duplicate-message throttle 206
no isolate 566
no monitor session 383, 408
no shutdown 566
no snmp trap link-status 273
no snmp-server protocol enable 276
no switch-profile 33
no system interface shutdown 566
no system mode maintenance 578
no system mode maintenance dont-generate-profile 578
no system mode maintenance on-reload reset-reason 576
ntp access-group {peer | serve | serve-only | query-only} 138
ntp authenticate 137
ntp authentication-key 136
ntp logging 140
ntp master 133
ntp peer 134
ntp server 134
ntp source 139
ntp source-interface 140
ntp trusted-key 136

O

option exporter-stats 462
option interface-table 462

P

periodic-inventory notification 205
periodic-inventory notification interval 205
periodic-inventory notification timeofday 205
permit 227, 516
phone-contact 194
ptp 86, 89
ptp device-type boundary-clock 81, 85
ptp device-type generalized-ptp 81, 85
ptp device-type ordinary-clock-grandmaster 81, 85
ptp scale on 1G 81
Ptp utc-offset 85
python instance 567, 571

R

record 463
reload 391, 418
rmon alarm 288
rmon event 289
rmon hcalarm 288
rollback running-config {checkpoint | file} 629

S

save 227
scheduler job name 236
scheduler logfile size 234
sflow collector-ip 505
show callhome 192, 195, 203, 205
show callhome destination-profile 192, 196, 198–199
show callhome destination-profile profile 196, 198–199
show callhome transport 192, 202, 204
show callhome user-def-cmds 192, 200
show cdp all 152
show cdp entry {all | name} 152
show cdp global 152
show cdp interface 152–153
show cdp neighbors {device-id | interface} 152
show cdp neighbors detail 150
show checkpoint 628–629
show checkpoint all 628
show checkpoint all system 628
show checkpoint all user 628
show checkpoint summary 628
show checkpoint summary system 628
show checkpoint summary user 628
show clock 587
show configuration session 227–229
show configuration session status 229
show configuration session summary 229
show diagnostic bootup level 301, 304
show diagnostic content module 302, 304
show diagnostic description module 304
show diagnostic events 304
show diagnostic ondemand setting 304
show diagnostic result module 305
show diagnostic simulation module 305
show diagnostic status module 303, 305
show diff rollback-patch {checkpoint | running-config | startup-config | file} 628–629
show event manager environment 314, 327
show event manager environment all 314, 327
show event manager event-types 327
show event manager event-types all 327
show event manager event-types module 327
show event manager history events 327
show event manager policy-state 316, 324, 327
show event manager script system 327
show event manager script system all 327
show event manager system-policy 308, 314, 327
show event manager system-policy all 327
show flow cache 452, 455, 480
show flow event 480

show flow exporter 452, 480
show flow filter 480
show flow interface 452, 480
show flow monitor 480
show flow profile 480
show flow record 452, 459, 480
show flow record netflow layer2-switched input 452, 466
show flow system 480
show hardware capacity 305
show install active 587, 592
show install inactive 592
show install log 592, 605
show install log detail 605
show install packages 603
show interface brief 580
show interface snmp-ifindex 273, 279
show ip access-list acl_nam 555
show ip access-lists 516, 520
show lldp interface 431–432, 439
show lldp neighbors detail 431
show lldp neighbors interface 431
show lldp timers 431, 440
show lldp tlv-select 431, 440
show lldp traffic 431
show lldp traffic interface 431
show logging console 160, 178
show logging info 163, 178
show logging last 176, 178
show logging level 165, 178
show logging logfile 176, 178
show logging logfile end-time 176, 178
show logging logfile start-time 176, 178
show logging module 164, 178
show logging monitor 161, 178
show logging nvram 176, 178
show logging nvram last 176, 178
show logging onboard 364
show logging onboard boot-uptime 364
show logging onboard counter-stats 364
show logging onboard credit-loss 364
show logging onboard device-version 364
show logging onboard endtime 364
show logging onboard environmental-history 364
show logging onboard error-stats 364
show logging onboard exception-log 364
show logging onboard interrupt-stats 364
show logging onboard module 364
show logging onboard obfl-history 365
show logging onboard obfl-logs 365
show logging onboard stack-trace 365
show logging onboard starttime 365
show logging onboard status 365
show logging origin-id 161
show logging server 168, 170, 178
show logging timestamp 166, 178
show mac access-lists 516, 520
show maintenance on-reload reset-reasons 580
show maintenance profile 580
show maintenance profile maintenance-mode 580
show maintenance profile normal-mode 580
show maintenance timeout 580
show module 305
show monitor session 390, 397, 411, 417, 421
show monitor session all 397, 411, 417
show monitor session range 397, 411, 417
show ntp access-groups 132, 139
show ntp authentication-keys 132, 136
show ntp authentication-status 132, 137
show ntp logging-status 132, 140
show ntp peer-status 132
show ntp peers 132, 135
show ntp rts-update 132
show ntp source 132
show ntp source-interface 132
show ntp statistics {io | local | memory | peer {ipaddr | name}} 132
show ntp trusted-keys 132, 136
show process 501
show ptp brief 110
show ptp clock 110
show ptp clock foreign-masters-record 110
show ptp corrections 110
show ptp counters 110
show ptp parent 110
show ptp port interface ethernet 110
show ptp time-property 110
show qos dcbxp interface 431
show rmon {alarms | hcalarms} 289
show rmon alarms 287
show rmon events 287, 289
show rmon hcalarms 287
show rmon logs 287
show rollback log 628
show rollback log exec 628
show rollback log verify 628
show run aclmgr 555
show run ofm 555
show running-config | include "scheduler aaa-authentication" 235
show running-config | include "system memory" 325
show running-config callhome 192
show running-config eem 308, 327
show running-config lldp 430–431
show running-config mmode 580
show running-config monitor 411
show running-config netflow 452, 455, 480
show running-config ntp 132–133
show running-config ptp 110
show running-config sflow 501
show running-config sflow all 501
show running-config snmp 279
show scheduler config 234, 239–240
show scheduler job 240
show scheduler logfile 240

show scheduler schedule 240
show sflow 501–502
show snapshots 580
show snapshots compare 581
show snapshots dump 581
show snapshots sections 574, 581
show snmp 279
show snmp community 280
show snmp context 276, 280
show snmp engineID 280
show snmp group 280
show snmp host 264, 280
show snmp session 280
show snmp source-interface 261, 264, 280
show snmp trap 280
show snmp user 255, 280
show startup-config callhome 192
show startup-config eem 327
show startup-config mmode 580
show startup-config monitor 411
show switch-profile 28, 30, 32
show system mode 577, 579, 581
show tech-support callhome 192
show tunnel-profile 555
shut 396
shutdown 566
site-id 195
sleep instance 566
snapshot delete 573
snapshot section add 574
snapshot section delete 574
snmp-server aaa-user cache-timeout 278
snmp-server community 257–258
snmp-server contact 194, 274
snmp-server counter cache timeout 277
snmp-server enable traps 267
snmp-server enable traps aaa 267
snmp-server enable traps bgp 267
snmp-server enable traps bridge 267
snmp-server enable traps callhome 268
snmp-server enable traps config 268
snmp-server enable traps eigrp 268
snmp-server enable traps entity 268
snmp-server enable traps feature-control 269
snmp-server enable traps hsrp 269
snmp-server enable traps license 269
snmp-server enable traps link 270
snmp-server enable traps ospf 270
snmp-server enable traps rf 270
snmp-server enable traps rmon 271
snmp-server enable traps snmp 271
snmp-server enable traps stpx 271
snmp-server enable traps syslog 271
snmp-server enable traps sysmgr 271
snmp-server enable traps upgrade 272
snmp-server enable traps vtp 272
snmp-server globalEnforcePriv 256
snmp-server host 259–260, 262, 264
snmp-server location 274
snmp-server name 254
snmp-server source-interface {traps | informs} 261
snmp-server tcp-session 274
snmp-server user 256–257, 261
source 461
source interface 390, 420
source-interface 210
statistics per-entry 516
storm-control action trap 272
streetaddress 195
switch-priority 195
switch-profile 27, 29, 31
switchport 28, 466–467, 469
sync-peers destination 28
system interface shutdown 566
system memory-thresholds minor 325
system memory-thresholds threshold critical no-process-kill 325
system mode maintenance dont-generate-profile 576
system mode maintenance on-reload reset-reason 576
system mode maintenance shutdown 576
system mode maintenance timeout 576

T

tag 315
template data timeout 462
terminal event-manager bypass 322
terminal monitor 159
time monthly 239
transport email from 202
transport email from callhome_email-address 209
transport email mail-server 201
transport email reply-to 202
transport email smtp-server hostname/ip-address port 465/587 use-vrf vrf-name 209
transport email username passwd {cleartext|encrypted} 209
transport http proxy enable 204
transport http proxy server 204
transport http use-vrf 203, 209
transport udp 461

U

udf 417

V

verify 30
version 9 462
vlan configuration 464