Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide, Release 10.6(x)

First Published: 2025-08-13

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 Cisco Systems, Inc. All rights reserved.

CONTENTS

Trademarks ?

PREFACE

Preface xiii

Audience xiii

Document Conventions xiii

Related Documentation for Cisco Nexus 9000 Series Switches xiv

Documentation Feedback xiv

Communications, services, and additional information xiv

Cisco Bug Search Tool xv

Documentation feedback xv

CHAPTER 1

New and Changed Information 1

New and Changed Information 1

CHAPTER 2

Overview 3

Licensing Requirements 3

Supported Platforms 3

Layer 2 Ethernet Switching Overview 3

VLANs 4

Spanning Tree 4

STP Overview 4

Rapid PVST+ 5

MST 5

STP Extensions 5

Traffic Storm Control 6

Related Topics 6

CHAPTER 3 **Configuring Layer 2 Switching** 7

Information About Layer 2 Switching 7

Layer 2 Ethernet Switching Overview 8

Switching Frames Between Segments 8

Building the Address Table and Address Table Changes 8

Consistent MAC Address Tables on the Supervisor and on the Modules 9

High Availability for Switching 9

Guidelines and limitations for the Cisco Nexus 93C64E-SG2-Q switch 9

Guidelines and limitations for Cisco Nexus 9336C-SE1 Cisco Nexus 9324C-SE1U and Cisco Nexus 9348Y2C6D-SE1U switches 10

Guidelines and limitations for Cisco Nexus 9396T12C-SE1 switch 11

Prerequisites for Configuring MAC Addresses 11

Default Settings for Layer 2 Switching 11

MAC Move Loop Detection 12

Generating Syslog Error Messages 12

Configuring Layer 2 Switching by Steps 13

Configuring a Static MAC Address 13

Disabling MAC Address Learning on System 15

Disabling MAC Address Learning on Layer 2 Interfaces 15

Disabling MAC Learning Per VLAN 16

Configuring the Aging Time for the MAC Table 18

Checking Consistency of MAC Address Tables 19

Clearing Dynamic Addresses from the MAC Table 20

Configuring Dynamic MAC Address Limits Per VLAN 21

Configuring L2 Heavy Mode 22

Verifying the Layer 2 Switching Configuration 23

Configuration Example for Layer 2 Switching 24

Additional References for Layer 2 Switching -- CLI Version 25

CHAPTER 4 **Configuring Flex Links** 27

Information About Flex Links 27

Flex Links 27

Preemption 28

Multicast 28
Guidelines and Limitations 28
Default Settings 30
Configuring Flex Links 30
Configuring Flexlinks 30
Configuring Flex Link Preemption 32
Verifying Configuration 34

CHAPTER 5 **Configuring VLANs 39**
Information About VLANs 39
Understanding VLANs 39
VLAN Ranges 40
About Reserved VLANs 41
Example of VLAN Reserve 42
Creating, Deleting, and Modifying VLANs 42
High Availability for VLANs 43
Prerequisites for Configuring VLANs 43
Guidelines and Limitations for Configuring VLANs 44
Default Settings for VLANs 45
Configuring a VLAN 45
Creating and Deleting a VLAN - CLI Version 45
Entering the VLAN Configuration Submode 47
Configuring a VLAN 48
Configuring a VLAN Before Creating the VLAN 50
Enabling the VLAN Long-Name 51
Configuring Inner VLAN and Outer VLAN Mapping on a Trunk Port 52
Verifying the VLAN Configuration 54
Displaying and Clearing VLAN Statistics 55
Configuration Example for VLANs 55
Additional References for VLANs 55

CHAPTER 6 **Configuring VTP 57**
Information About VTP 57
VTP 57

VTP Overview 58
VTP Modes 58
VTP Per Interface 58
Guidelines and Limitations for Configuring VTP 58
Default Settings 59
Configuring VTP 59

CHAPTER 7 **Configuring Private VLANs Using NX-OS 63**
Information About Private VLANs 63
Private VLAN Overview 64
Primary and Secondary VLANs in Private VLANs 64
Private VLAN Ports 64
Primary, Isolated, and Community Private VLANs 66
Associating Primary and Secondary VLANs 67
Broadcast Traffic in Private VLANs 68
Private VLAN Port Isolation 68
Private VLANs and VLAN Interfaces 69
Private VLANs Across Multiple Devices 69
Private VLAN with Inner VLAN Tag Preservation 69
High Availability for Private VLANs 70
Prerequisites for Private VLANs 71
Guidelines and Limitations for Configuring Private VLANs 71
Default Settings for Private VLANs 74
Configuring a Private VLAN 74
Enabling Private VLANs - CLI Version 75
Configuring a VLAN as a Private VLAN - CLI Version 76
Associating Secondary VLANs with a Primary Private VLAN - CLI Version 77
Mapping Secondary VLANs to the VLAN Interface of a Primary VLAN - CLI Version 79
Configuring a Layer 2 Interface as a Private VLAN Host Port 81
Configuring a Layer 2 Interface as a Private VLAN Isolated Trunk Port 83
Configuring a Layer 2 Interface as a Private VLAN Promiscuous Port 85
Configuring a Layer 2 Interface as a Private VLAN Promiscuous Trunk Port 87
Verifying the Private VLAN Configuration 90
Displaying and Clearing Private VLAN Statistics 90

Configuration Examples for Private VLANs 90
Additional References for Private VLANs -- CLI Version 91

CHAPTER 8 Configuring Switching Modes 93
Information About Switching Modes 93
Guidelines and Limitations for Switching Modes 93
Default Settings for Switching Modes 94
Configuring Switching Modes 94
Enabling Store-and-Forward Switching 94
Reenabling Cut-Through Switching 95

CHAPTER 9 Configuring Rapid PVST+ Using Cisco NX-OS 97
Information About Rapid PVST+ 97
STP 98
Overview of STP 98
How a Topology is Created 98
Bridge ID 99
BPDUs 101
Election of the Root Bridge 101
Creating the Spanning Tree Topology 102
Rapid PVST+ 102
Overview of Rapid PVST+ 102
Rapid PVST+ BPDUs 104
Proposal and Agreement Handshake 105
Protocol Timers 106
Port Roles 106
Rapid PVST+ Port State Overview 107
Synchronization of Port Roles 109
Detecting Unidirectional Link Failure:Rapid PVST+ 110
Port Cost 111
Port Priority 111
Rapid PVST+ and IEEE 802.1Q Trunks 112
Rapid PVST+ Interoperation with Legacy 802.1D STP 112
Rapid PVST+ Interoperation with 802.1s MST 113

High Availability for Rapid PVST+ 113
Prerequisites for Configuring Rapid PVST+ 113
Guidelines and Limitations for Configuring Rapid PVST+ 113
Default Settings for Rapid PVST+ 114
Configuring Rapid PVST+ 115
Enabling Rapid PVST+ - CLI Version 115
Disabling or Enabling Rapid PVST+ Per VLAN - CLI Version 117
Configuring the Root Bridge ID 119
Configuring a Secondary Root Bridge-CLI Version 120
Configuring the Rapid PVST+ Bridge Priority of a VLAN 121
Configuring the Rapid PVST+ Port Priority - CLI Version 123
Configuring the Rapid PVST+ Path-Cost Method and Port Cost - CLI Version 124
Configuring the Rapid PVST+ Hello Time for a VLAN - CLI Version 126
Configuring the Rapid PVST+ Forward Delay Time for a VLAN - CLI Version 127
Configuring the Rapid PVST+ Maximum Age Time for a VLAN - CLI Version 128
Specifying the Link Type for Rapid PVST+ - CLI Version 129
Reinitializing the Protocol for Rapid PVST+ 131
Verifying the Rapid PVST+ Configurations 131
Displaying and Clearing Rapid PVST+ Statistics -- CLI Version 132
Rapid PVST+ Example Configurations 132
Additional References for Rapid PVST+ -- CLI Version 132

CHAPTER 10 **Configuring MST Using Cisco NX-OS 135**
Information About MST 135
MST Overview 136
MST Regions 136
MST BPDUs 137
MST Configuration Information 137
IST, CIST, and CST 138
IST, CIST, and CST Overview 138
Spanning Tree Operation Within an MST Region 138
Spanning Tree Operations Between MST Regions 139
MST Terminology 139
Hop Count 140

Boundary Ports 140
Detecting Unidirectional Link Failure: MST 141
Port Cost and Port Priority 141
Interoperability with IEEE 802.1D 142
High Availability for MST 142
Prerequisites for MST 143
Guidelines and Limitations for Configuring MST 143
Default Settings for MST 144
Configuring MST 145
Enabling MST - CLI Version 145
Entering MST Configuration Mode 147
Specifying the MST Name 148
Specifying the MST Configuration Revision Number 149
Specifying the Configuration on an MST Region 151
Mapping or Unmapping a VLAN to an MST Instance - CLI Version 153
Configuring the Root Bridge 154
Configuring an MST Secondary Root Bridge 157
Configuring the MST Switch Priority 158
Configuring the MST Port Priority 160
Configuring the MST Port Cost 161
Configuring the MST Hello Time 163
Configuring the MST Forwarding-Delay Time 164
Configuring the MST Maximum-Aging Time 165
Configuring the MST Maximum-Hop Count 166
Configuring an Interface to Proactively Send Prestandard MSTP Messages - CLI Version 167
Specifying the Link Type for MST - CLI Version 169
Reinitializing the Protocol for MST 170
Verifying the MST Configuration 171
Displaying and Clearing MST Statistics -- CLI Version 171
MST Example Configuration 172
Additional References for MST -- CLI Version 173

CHAPTER 11 **Configuring STP Extensions Using Cisco NX-OS** 175
Information About STP Extensions 175

STP Port Types 176
STP Edge Ports 176
Bridge Assurance 176
BPDU Guard 178
BPDU Filtering 178
Loop Guard 179
Root Guard 180
Applying STP Extension Features 180
PVST Simulation 180
High Availability for STP 181
Prerequisites for STP Extensions 181
Guidelines and Limitations for Configuring STP Extensions 181
Default Settings for STP Extensions 183
Configuring STP Extensions Steps 183
Configuring Spanning Tree Port Types Globally 183
Configuring Spanning Tree Edge Ports on Specified Interfaces 185
Configuring Spanning Tree Network Ports on Specified Interfaces 187
Enabling BPDU Guard Globally 189
Enabling BPDU Guard on Specified Interfaces 190
Enabling BPDU Filtering Globally 192
Enabling BPDU Filtering on Specified Interfaces 194
Enabling Loop Guard Globally 196
Enabling Loop Guard or Root Guard on Specified Interfaces 197
Configuring PVST Simulation Globally-CLI Version 199
Configuring PVST Simulation Per Port 201
Verifying the STP Extension Configuration 202
Configuration Examples for STP Extension 203
Additional References for STP Extensions -- CLI Version 203

CHAPTER 12 **Configuring Reflective Relay for Layer 2 Switching** 205
About Reflective Relay 802.1Qbg 205
Reflective Relay Support 205
Guidelines and Limitations for Reflective Relay 205
Configuring Reflective Relay Using the NX-OS CLI 206

CHAPTER 13 **Configuring Link-State Tracking** 207

Link-State Tracking 207

How Link-State Tracking Works 207

Guidelines and Limitation for Link-State Tracking 208

Configure Link-State Tracking 209

Monitor the Link-State Tracking 210

Link-State Tracking Configuration Examples 212

Preface

This preface includes the following sections:

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

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

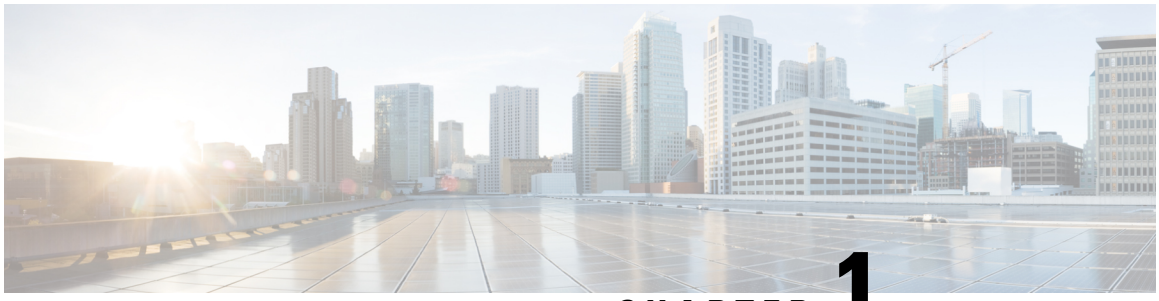
- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class 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

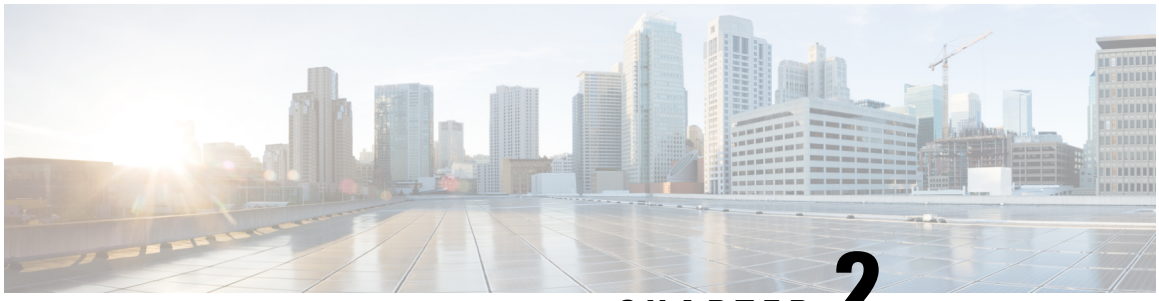
New and Changed Information

This table summarizes the new and changed features for the *Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide, Release 10.6(x)* and where they are documented.

Table 1: New and Changed Features

Feature	Description	Changed in Release	Where Documented
Support for Layer 2 switching on 9800 EoR line cards	Added support for Layer 2 switching features on Cisco Nexus 9800 Series switches with N9K-X9836DM-A and N9K-X98900CD-A line cards.	10.6(3)F	Guidelines and limitations for the Cisco Nexus 9800 Series switches, on page 10
Static mac and Learn Disable are supported on Nexus N9336C-SE1 and 9396T12C-SE1 switches.	Beginning with Cisco NX-OS Release 10.6(3)F, Static mac and Learn Disable Cisco Nexus N9336C-SE1 and 9396T12C-SE1 platform switches.	10.6(3)F	Guidelines and limitations for Cisco Nexus 9336C-SE1 switch Guidelines and limitations for Cisco Nexus 9396T12C-SE1 switch
UDLD	Beginning with Cisco NX-OS Release 10.6(3)F, UDLD is supported on Nexus 9300-SE1 switches.	10.6(3)F	Guidelines and limitations for the Cisco Nexus 93C64E-SG2-Q switch
QinQ multi tag support on Nexus N9336C-SE1 and 9396T12C-SE1	Beginning with Cisco NX-OS Release 10.6(3)F, QinQ Multi Tag is supported on Nexus N9336C-SE1 and 9396T12C-SE1 switches.	10.6(3)F	Guidelines and limitations for the Cisco Nexus 93C64E-SG2-Q switch

Feature	Description	Changed in Release	Where Documented
Support SVI Autostate on Nexus N9336C-SE1 and 9396T12C-SE1	Beginning with Cisco NX-OS Release 10.6(3)F,Support SVI Autostate on S1 platforms is supported on Nexus N9336C-SE1 and 9396T12C-SE1 switches.	10.6(3)F	Guidelines and limitations for the Cisco Nexus 93C64E-SG2-Q switch
Link-State Tracking	Added support for Link-State Tracking to bind multiple interface link states for redundancy and stability.	10.6(2)F	Link-State Tracking, on page 207
Support for Rapid PVST+ on Cisco N9324C-SE1U and Cisco N9348Y2C6D-SE1U switches	Added support for Rapid PVST+ spanning tree protocol on Cisco N9324C-SE1U and N9348Y2C6D-SE1U switches.	10.6(2)F	Guidelines and Limitations on N9300-Series Smart Switches
Support for MST on Cisco N9324C-SE1U and Cisco N9348Y2C6D-SE1U switches	Added support for Multiple Spanning Tree (MST) protocol on Cisco N9324C-SE1U and N9348Y2C6D-SE1U switches.	10.6(2)F	Guidelines and Limitations on N9300-Series Smart Switches
Support for Layer 2 switching features on Cisco N9324C-SE1U and Cisco N9348Y2C6D-SE1U switches	Added support for Layer 2 switching features.	10.6(2)F	Guidelines and limitations for Cisco Nexus 9336C-SE1 Cisco Nexus 9324C-SE1U and Cisco Nexus 9348Y2C6D-SE1U switches, on page 10
Support for Layer 2 switching features on Cisco N9336C-SE1 switch.	Added support for Layer 2 switching features.	10.6(1)F	Guidelines and limitations for Cisco Nexus 9336C-SE1 Cisco Nexus 9324C-SE1U and Cisco Nexus 9348Y2C6D-SE1U switches, on page 10



CHAPTER 2

Overview

- Licensing Requirements, on page 3
- Supported Platforms, on page 3
- Layer 2 Ethernet Switching Overview, on page 3
- VLANs, on page 4
- Spanning Tree , on page 4
- Traffic Storm Control, on page 6
- Related Topics, on page 6

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.

Layer 2 Ethernet Switching Overview

The device supports simultaneous, parallel connections between Layer 2 Ethernet segments. Switched connections between Ethernet segments last only for the duration of the packet. New connections can be made between different segments for the next packet.

The device solves congestion problems caused by high-bandwidth devices and a large number of users by assigning each device (for example, a server) to its own collision domain. Because each LAN port connects to a separate Ethernet collision domain, servers in a switched environment achieve full access to the bandwidth.

Because collisions cause significant congestion in Ethernet networks, an effective solution is full-duplex communication. Typically, 10/100-Mbps Ethernet operates in half-duplex mode, which means that stations can either receive or transmit. In full-duplex mode, which is configurable on these interfaces, two stations can transmit and receive at the same time. When packets can flow in both directions simultaneously, the effective Ethernet bandwidth doubles.

VLANs

A VLAN is a switched network that is logically segmented by function, project team, or application, without regard to the physical locations of the users. VLANs have the same attributes as physical LANs, but you can group end stations even if they are not physically located on the same LAN segment.

Any switch port can belong to a VLAN, and unicast, broadcast, and multicast packets are forwarded and flooded only to end stations in that VLAN. Each VLAN is considered as a logical network, and packets destined for stations that do not belong to the VLAN must be forwarded through a bridge or a router.

All ports are assigned to the default VLAN (VLAN1) when the device first comes up. A VLAN interface, or switched virtual interface (SVI), is a Layer 3 interface that is created to provide communication between VLANs.

The devices support 4095 VLANs in accordance with the IEEE 802.1Q standard. These VLANs are organized into several ranges, and you use each range slightly differently. Some of these VLANs are reserved for internal use by the device and are not available for configuration.

Note Inter-Switch Link (ISL) trunking is not supported on the Cisco NX-OS.

Spanning Tree

This section discusses the implementation of the Spanning Tree Protocol (STP) on the software. Spanning tree is used to refer to IEEE 802.1w and IEEE 802.1s. When the IEEE 802.1D Spanning Tree Protocol is referred to in the publication, 802.1D is stated specifically.

STP Overview

STP provides a loop-free network at the Layer 2 level. Layer 2 LAN ports send and receive STP frames, which are called Bridge Protocol Data Units (BPDUs), at regular intervals. Network devices do not forward these frames but use the frames to construct a loop-free path.

802.1D is the original standard for STP, and many improvements have enhanced the basic loop-free STP. You can create a separate loop-free path for each VLAN, which is named Per VLAN Spanning Tree (PVST+). Additionally, the entire standard was reworked to make the loop-free convergence process faster to keep up with the faster equipment. This STP standard with faster convergence is the 802.1w standard, which is known as Rapid Spanning Tree (RSTP). Now, these faster convergence times are available as you create STP for each VLAN, which is known as Per VLAN Rapid Spanning Tree (Rapid PVST+).

Finally, the 802.1s standard, Multiple Spanning Trees (MST), allows you to map multiple VLANs into a single spanning tree instance. Each instance runs an independent spanning tree topology.

Although the software can interoperate with legacy 802.1D systems, the system runs Rapid PVST+ and MST. Rapid PVST+ is the default STP protocol for Cisco Nexus devices.

Note Cisco NX-OS uses the extended system ID and MAC address reduction; you cannot disable these features.

In addition, Cisco has created some proprietary features to enhance the spanning tree activities.

Rapid PVST+

Rapid PVST+ is the default spanning tree mode for the software and is enabled by default on the default VLAN and all newly created VLANs.

A single instance, or topology, of RSTP runs on each configured VLAN, and each Rapid PVST+ instance on a VLAN has a single root device. You can enable and disable STP on a per-VLAN basis when you are running Rapid PVST+.

MST

The software also supports MST. The multiple independent spanning tree topologies enabled by MST provide multiple forwarding paths for data traffic, enable load balancing, and reduce the number of STP instances required to support a large number of VLANs.

MST incorporates RSTP, so it also allows rapid convergence. MST improves the fault tolerance of the network because a failure in one instance (forwarding path) does not affect other instances (forwarding paths).

Note Changing the spanning tree mode disrupts the traffic because all spanning tree instances are stopped for the previous mode and started for the new mode.

You can force specified interfaces to send prestandard, rather than standard, MST messages using the command-line interface.

STP Extensions

The software supports the following Cisco proprietary features:

- Spanning tree port types—The default spanning tree port type is normal. You can configure interfaces connected to Layer 2 hosts as edge ports and interfaces connected to Layer 2 switches or bridges as network ports.
- Bridge Assurance—Once you configure a port as a network port, Bridge Assurance sends BPDUs on all ports and moves a port into the blocking state if it no longer receives BPDUs. This enhancement is available only when you are running Rapid PVST+ or MST.
- BPDU Guard—BPDU Guard shuts down the port if that port receives a BPDU.
- BPDU Filter—BPDU Filter suppresses sending and receiving BPDUs on the port.
- Loop Guard—Loop Guard helps prevent bridging loops that could occur because of a unidirectional link failure on a point-to-point link.
- Root Guard—STP root guard prevents a port from becoming root port or blocked port. If a port configured for root guard receives a superior BPDU, the port immediately goes to the root-inconsistent (blocked) state.

Traffic Storm Control

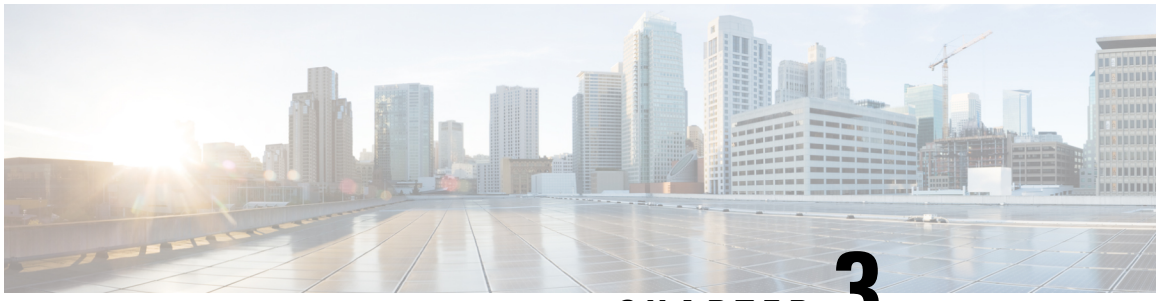
Traffic storm control (also called traffic suppression) allows you to monitor the levels of the incoming traffic over a 1-second interval. During this interval, the traffic level, which is a percentage of the total available bandwidth of the port, is compared with the traffic storm control level that you configured. When the ingress traffic reaches the traffic storm control level that is configured on the port, traffic storm control drops the traffic until the interval ends.

For more information, see the Configuring Traffic Storm Control chapter in the Cisco Nexus 9000 Series NX-OS Security Configuration Guide. Beginning with Cisco NX-OS Release 10.3(2)F, the traffic storm control feature is also supported on Layer 3.

Related Topics

The following documents are related to the Layer 2 switching features:

- *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*
- *Cisco Nexus 9000 Series NX-OS Security Configuration Guide*
- *Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide*
- *Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*



CHAPTER 3

Configuring Layer 2 Switching

- Information About Layer 2 Switching, on page 7
- High Availability for Switching, on page 9
- Guidelines and limitations for the Cisco Nexus 93C64E-SG2-Q switch , on page 9
- Guidelines and limitations for Cisco Nexus 9336C-SE1 Cisco Nexus 9324C-SE1U and Cisco Nexus 9348Y2C6D-SE1U switches, on page 10
- Guidelines and limitations for Cisco Nexus 9396T12C-SE1 switch, on page 11
- Prerequisites for Configuring MAC Addresses, on page 11
- Default Settings for Layer 2 Switching, on page 11
- MAC Move Loop Detection, on page 12
- Generating Syslog Error Messages, on page 12
- Configuring Layer 2 Switching by Steps, on page 13
- Verifying the Layer 2 Switching Configuration, on page 23
- Configuration Example for Layer 2 Switching, on page 24
- Additional References for Layer 2 Switching -- CLI Version, on page 25

Information About Layer 2 Switching

Note See the Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide, for information on creating interfaces.

You can configure Layer 2 switching ports as access or trunk ports. Trunks carry the traffic of multiple VLANs over a single link and allow you to extend VLANs across an entire network. All Layer 2 switching ports maintain MAC address tables.

Note See the Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide, for complete information on high-availability features.

Layer 2 Ethernet Switching Overview

The device supports simultaneous, parallel connections between Layer 2 Ethernet segments. Switched connections between Ethernet segments last only for the duration of the packet. New connections can be made between different segments for the next packet.

The device solves congestion problems caused by high-bandwidth devices and a large number of users by assigning each device (for example, a server) to its own collision domain. Because each LAN port connects to a separate Ethernet collision domain, servers in a switched environment achieve full access to the bandwidth.

Because collisions cause significant congestion in Ethernet networks, an effective solution is full-duplex communication. Typically, 10/100-Mbps Ethernet operates in half-duplex mode, which means that stations can either receive or transmit. In full-duplex mode, which is configurable on these interfaces, two stations can transmit and receive at the same time. When packets can flow in both directions simultaneously, the effective Ethernet bandwidth doubles.

Switching Frames Between Segments

Each LAN port on a device can connect to a single workstation, server, or to another device through which workstations or servers connect to the network.

To reduce signal degradation, the device considers each LAN port to be an individual segment. When stations connected to different LAN ports need to communicate, the device forwards frames from one LAN port to the other at wire speed to ensure that each session receives full bandwidth.

To switch frames between LAN ports efficiently, the device maintains an address table. When a frame enters the device, it associates the media access control (MAC) address of the sending network device with the LAN port on which it was received.

Building the Address Table and Address Table Changes

The device dynamically builds the address table by using the MAC source address of the frames received. When the device receives a frame for a MAC destination address not listed in its address table, it floods the frame to all LAN ports of the same VLAN except the port that received the frame. When the destination station replies, the device adds its relevant MAC source address and port ID to the address table. The device then forwards subsequent frames to a single LAN port without flooding all LAN ports.

You can configure MAC addresses, which are called static MAC addresses, to statically point to specified interfaces on the device. These static MAC addresses override any dynamically learned MAC addresses on those interfaces. You cannot configure broadcast addresses as static MAC addresses. The static MAC entries are retained across a reboot of the device.

You must manually configure identical static MAC addresses on both devices connected by a virtual port channel (vPC) peer link. The MAC address table display is enhanced to display information on MAC addresses when you are using vPCs.

See the Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide* for information about vPCs.

The address table can store a number of MAC address entries depending on the hardware I/O module. The device uses an aging mechanism, defined by a configurable aging timer, so if an address remains inactive for a specified number of seconds, it is removed from the address table.

Consistent MAC Address Tables on the Supervisor and on the Modules

Optimally, all the MAC address tables on each module exactly match the MAC address table on the supervisor. When you enter the **show forwarding consistency l2** command or the **show consistency-checker l2** command, the device displays discrepant, missing, and extra MAC address entries.

High Availability for Switching

You can upgrade or downgrade the software seamlessly, with respect to classical Ethernet switching. If you have configured static MAC addresses on Layer 3 interfaces, you must unconfigure those ports in order to downgrade the software.

Note See the *Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide*Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide, for complete information on high availability features.

Guidelines and limitations for the Cisco Nexus 93C64E-SG2-Q switch

Starting with Cisco NX-OS Release 10.5(3)F, the Cisco Nexus 93C64E-SG2-Q switch supports only these Layer 2 switching features.

- VLAN
- STP
- Layer 2 Consistency Checker
- MAC learning

These guidelines are applicable to MAC learning on the Cisco Nexus 93C64E-SG2-Q switch.

- The **switchport mac-learn disable** configuration s *not* supported at the interface level.
- **mac learn disable** configuration is *not* supported at the interface level and VLAN level.

Static MAC address is *not* supported on the Cisco Nexus 93C64E-SG2-Q switch.

Starting with Cisco NX-OS Release 10.6(3)F, the Cisco Nexus 9300-SE1 and Cisco Nexus 9364E-SG2 supports these Layer 2 features.

- Unidirectional Link Detection (UDLD)
- Switch virtual interface (SVI) and VLAN logical interfaces
- Q-in-Q VLAN Tunnels

Guidelines and limitations for the Cisco Nexus 9800 Series switches

Starting with Cisco NX-OS Release 10.6(3)F, the Cisco Nexus 9800 Series switches equipped with N9K-X9836DM-A and N9K-X98900CD-A line cards supports only these Layer 2 switching features.

- VLAN
- STP
- Rapid PVST+
- MST
- MAC learning

Unsupported features:

- Static MAC address
- Disable MAC learning globally or at Layer 2 interfaces or per VLAN
- MAC Move Loop Detection
- Reflective relay
- Flex links
- VTP
- Private VLAN
- Storm traffic control

Guidelines and limitations for Cisco Nexus 9336C-SE1 Cisco Nexus 9324C-SE1U and Cisco Nexus 9348Y2C6D-SE1U switches

- Starting with Cisco NX-OS Release 10.6(2)F, the Cisco N9324C-SE1U and Cisco N9348Y2C6D-SE1U switches support these Layer 2 switching features.
  - VLAN
  - Spanning Tree Protocol (STP)
  - Rapid PVST+
  - MST
  - MAC learning (global)
- Starting with Cisco NX-OS Release 10.6(1)F, the Cisco Nexus 9336C-SE1 supports Layer 2 switching features mentioned above.
- These features are *not* supported in Cisco NX-OS Release 10.6(1)F on the Cisco Nexus 9336C-SE1switch.
  - Static MAC address

- Disable MAC learning at Layer 2 interfaces or per VLAN
- Flex links
- VTP
- Private VLAN
- Storm traffic control
- Reflective relay

- Starting with Cisco NX-OS Release 10.6(3)F, the Cisco Nexus 9336C-SE1 switches support these Layer 2 switching features.
  - Static MAC address
  - Disable MAC learning at Layer 2 per VLAN

Guidelines and limitations for Cisco Nexus 9396T12C-SE1 switch

- Starting with Cisco NX-OS Release 10.6(3)F, the Cisco Nexus 9396T12C-SE1 switches support these Layer 2 switching features.
  - Static MAC address
  - Disable MAC learning at Layer 2 per VLAN

Prerequisites for Configuring MAC Addresses

MAC addresses have the following prerequisites:

- You must be logged onto the device.
- If necessary, install the Advanced Services license.

Default Settings for Layer 2 Switching

This table lists the default setting for Layer 2 switching parameters.

Table 2: Default Layer 2 Switching Parameters

Parameters	Default
Aging time	1800 seconds

MAC Move Loop Detection

Cisco Nexus 9000 Series switches leverage L2FM for software MAC learning (and, subsequently, loop detection). If a host (MAC address) moves between two interfaces within the same VLAN, it would trigger a MAC move. If there are a large number of such MAC moves in a short duration of time, the control plane of the switch and the CPU performance could get impacted. L2FM protects the switch from such scenarios by disabling MAC learning on the specific VLAN once the number of MAC moves for the corresponding MAC address exceeds a threshold.

For Broadcom ASIC based switches, the MAC move learn disable threshold criteria is when a single MAC address moves 10 or more times in a duration of 1 second within the same VLAN.

For Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2/H2R/H1, 9804/9808 switches and Cisco Nexus 9500 switches with 9700-EX/FX/GX line cards, the MAC move learn disable threshold criteria is when a single MAC address moves 10 or more times in 10 seconds within the same VLAN.

Once the threshold limit is hit, all new MAC learning on the corresponding VLAN gets disabled for a period of 120 seconds. After 120 seconds, new MAC learning is re-enabled on that VLAN. There is no impact of this on the rest of the VLANs on the switch.

In Cisco NX-OS Release 10.5(3)F, Cisco Nexus 93C64E-SG2-Q switch does *not* support MAC Move Loop Detection feature.

Generating Syslog Error Messages

To see MAC move notifications in syslogs, follow the below steps:

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**logging level l2fm 5** **Example:** `switch(config)# logging level l2fm 5`	Enables logging of all L2FM events from level 5 up to the highest severity events.
Step 3	(Optional) **mac address-table notification mac-move** **Example:** `switch(config)# mac address-table notification mac-move`	Enables MAC move notification on the switch. **Note** • MAC move notification is enabled by default. • This command ensures that the syslog for L2FM detect displays when there is a MAC address move.

Following are the sample generated syslog messages:

- When MAC move is detected:

```
2023 Nov 29 21:42:04 N-3164Q-40G %L2FM-4-L2FM_MAC_MOVE2: Mac
0003.0001.005d in vlan 500 has moved from Eth1/24 to Eth1/63
```

- When MAC learning on VLAN is disabled:

```
2023 Nov 29 21:23:29 N-3164Q-40G %L2FM-2-L2FM_MAC_FLAP_DISABLE_LEARN:
Disabling learning in vlan 500 for 120s due to too many mac moves
```

- When MAC learning on VLAN is re-enabled:

```
2023 Nov 29 21:23:19 N-3164Q-40G %L2FM-2-L2FM_MAC_FLAP_RE_ENABLE_LEARN:
Re-enabling learning in vlan 500
```

Example

In order to check if the MAC addresses move, enter the command:

```
switch# show mac address-table notification mac-move
MAC Move Notify Triggers: 1206
Number of MAC Addresses added: 944088
Number of MAC Addresses moved: 265
Number of MAC Addresses removed: 943920
```

Note The following are the possible causes for MAC moves:

- MAC addresses move because of server NIC teaming and moving between Active-Active, Active-Standby states, etc.
- MAC addresses move because the source of the data is physically moved across all switches while STP states are converged and in correct states.
- Due to loops in the network.

Configuring Layer 2 Switching by Steps

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Configuring a Static MAC Address

You can configure MAC addresses, which are called static MAC addresses, to statically point to specified interfaces on the device. These static MAC addresses override any dynamically learned MAC addresses on those interfaces. You cannot configure broadcast or multicast addresses as static MAC addresses.

SUMMARY STEPS

1. **config t**
2. **mac address-table static** *mac-address* **vlan** *vlan-id* {[**drop** | **interface** {*type slot/port*} | **port-channel** *number*]}
3. **exit**
4. (Optional) **show mac address-table static**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**mac address-table static** *mac-address* **vlan** *vlan-id* {[**drop** | **interface** {*type slot/port*} | **port-channel** *number*]} **Example:** `switch(config)# mac address-table static 1.1.1 vlan 2 interface ethernet 1/2`	Specifies a static MAC address to add to the Layer 2 MAC address table. **Note** Use the **drop** option to drop all traffic that is going to the configured MAC address in the specified VLAN. MAC static drop condition is ignored for routed traffic egressing the SVI corresponding to the VLAN where the mac static drop is configured. This issue only impacts routed traffic (MAC drop configuration in the vlan associated with outbound SVI). This does not apply to bridged traffic where traffic ingressing 9K egresses in the same VLAN (L2 forwarded).

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

Step 4	(Optional) **show mac address-table static** **Example:** `switch# show mac address-table static`	Displays the static MAC addresses.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to put a static entry in the Layer 2 MAC address table:

```
switch# config t
switch(config)# mac address-table static 1.1.1 vlan 2 interface ethernet 1/2
switch(config)#
```

Disabling MAC Address Learning on System

You can now disable and re-enable MAC address learning on system.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config-if)# **[no] mac-learn disable**
3. switch(config-if)# **clear mac address-table dynamic**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config-if)# **[no] mac-learn disable**	Disables MAC address learning on the switch. The **no** form of this command re-enables MAC address learning on the switch.
Step 3	switch(config-if)# **clear mac address-table dynamic**	Clears the MAC address table for the switch. **Important** After disabling MAC address learning on the switch, ensure that you clear the MAC address table.

Disabling MAC Address Learning on Layer 2 Interfaces

You can now disable and re-enable MAC address learning on Layer 2 interfaces.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot/port*
3. switch(config-if)# **[no] switchport mac-learn disable**
4. switch(config-if)# **clear mac address-table dynamic interface** *type slot/port*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot/port*	Enters the interface configuration mode for the specified interface.
Step 3	switch(config-if)# **[no] switchport mac-learn disable**	Disables MAC address learning on Layer 2 interfaces. The **no** form of this command re-enables MAC address learning on Layer 2 interfaces. **Note** In Warp mode, the Cisco Nexus 3500 switch does not flood Layer 3 traffic to the VLAN in which the port configured using **switchport mac-learn disable** is present, and the traffic is dropped. In Normal mode, the switch should flood the Layer 3 traffic to this VLAN.
Step 4	switch(config-if)# **clear mac address-table dynamic interface** *type slot/port*	Clears the MAC address table for the specified interface. **Important** After disabling MAC address learning on an interface, ensure that you clear the MAC address table.

Example

This example shows how to disable MAC address learning on Layer 2 interfaces:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# switchport mac-learn disable
switch(config-if)# clear mac address-table dynamic interface ethernet 1/4
```

This example shows how to re-enable MAC address learning on Layer 2 interfaces:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# no switchport mac-learn disable
```

Disabling MAC Learning Per VLAN

Beginning with Cisco NX-OS Release 10.5(1)F, MAC learning can be disabled at the VLAN level on the Cisco Nexus 9300-FX/FX2/FX3/GX/GX2/H2R/H1 Series switches and Cisco Nexus 9500 Series switches with 9700-EX/FX/GX line cards.

Whenever MAC learning is disabled on a VLAN, a syslog message will be generated to confirm that MAC learning is disabled. Also, a notification is sent to clear mac address that have already been learned on the VLAN, if necessary. The syslog will also advise applying the same configuration on the peer vPC.

Before you begin

- To use the mac-learn disable feature, ensure that VLANs are created.
- The VLAN can be a legacy VLAN or a VXLAN VLAN.

SUMMARY STEPS

1. **config t**
2. **[no] mac-learn vlan** *vlan-id*
3. **clear mac address-table dynamic vlan** *vlan-id*
4. **exit**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**[no] mac-learn vlan** *vlan-id*
	Example:
	`switch(config)# no mac-learn vlan 6`
		Disables the mac-learn on the specified VLAN.
		Use the **mac-learn vlan** *vlan-id* command, to enable the mac-learn on the specified VLAN.
		The allowed VLAN range is 2–4092, except for the reserved VLANs.
		Note
		• This configuration works on standalone and vPC. However, make sure to disable mac-learn on both the vPC peers.
		• This configuration is mutually exclusive with following features:
		  • SVI
		  • Port-security
		  • Private VLAN

Step 3	**clear mac address-table dynamic vlan** *vlan-id* **Example:** `switch(config)# clear mac address-table dynamic vlan 6`	Clears the MAC address table for the specified VLAN. **Important** After disabling MAC address learning on VLAN, ensure that you clear the MAC address table.

	Command or Action	Purpose

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

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

Configuring the Aging Time for the MAC Table

You can configure the amount of time that a MAC address entry (the packet source MAC address and port on which that packet was learned) remains in the MAC table, which contains the Layer 2 information.

Note When configuring the MAC aging timer, the observed aging time may exceed the configured timer due to internal mechanisms. This additional time accounts for the **refresh timer** that is approximately 60 seconds and is used to ensure all MAC addresses in the pending deletion list are properly cleared before final deletion. The **delta** represents a small additional buffer time due to internal processing delays.

For **port-channel interfaces**, the actual MAC aging time is calculated as:

```
2x (configured MAC aging timer) + 60 seconds (refresh timer) +
        delta
```

For **physical interfaces**, the MAC aging time occurs within:

```
`2x (configured MAC aging timer) + delta`
```

Note MAC addresses are aged out up to two times the configured MAC address table aging timeout.

Note You can also configure the MAC aging time in interface configuration mode or VLAN configuration mode.

SUMMARY STEPS

1. **config t**
2. **mac address-table aging-time** *seconds*
3. **exit**
4. (Optional) **show mac address-table aging-time**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**mac address-table aging-time** *seconds* **Example:** `switch(config)# mac address-table aging-time 600`	Specifies the time before an entry ages out and is discarded from the Layer 2 MAC address table. The range is from 120 to 918000; the default is 1800 seconds. Entering the value 0 disables the MAC aging.

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

Step 4	(Optional) **show mac address-table aging-time** **Example:** `switch# show mac address-table aging-time`	Displays the aging time configuration for MAC address retention.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to set the ageout time for entries in the Layer 2 MAC address table to 600 seconds (10 minutes):

```
switch# config t
switch(config)# mac address-table aging-time 600
switch(config)#
```

Checking Consistency of MAC Address Tables

You can check the match between the MAC address table on the supervisor and all the modules.

SUMMARY STEPS

1. **show consistency-checker l2 module** <*slot_number*>

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**show consistency-checker l2 module** <*slot_number*>
	Example:
	`switch# show consistency-checker l2 module 7`
	`switch#`
		Displays the discrepant, missing, and extra MAC addresses between the supervisor and the specified module.

Example

This example shows how to display discrepant, missing, and extra entries in the MAC address tables between the supervisor and the specified module:

```
switch# show consistency-checker l2 module 7
switch#
```

Clearing Dynamic Addresses from the MAC Table

You can clear all dynamic Layer 2 entries in the MAC address table. (You can also clear entries by designated interface or VLAN.)

SUMMARY STEPS

1. **clear mac address-table dynamic** {**address** *mac_addr*} {**interface** [**ethernet** *slot/port* | **port-channel** *channel-number*]} {**vlan** *vlan_id*}
2. (Optional) **show mac address-table**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**clear mac address-table dynamic** {**address** *mac_addr*} {**interface** [**ethernet** *slot/port*	**port-channel** *channel-number*]} {**vlan** *vlan_id*} **Example:** `switch# clear mac address-table dynamic`	Clears the dynamic address entries from the MAC address table in Layer 2.
Step 2	(Optional) **show mac address-table** **Example:** `switch# show mac address-table`	Displays the MAC address table.

Example

This example shows how to clear the dynamic entries in the Layer 2 MAC address table:

```
switch# clear mac address-table dynamic
switch#
```

Configuring Dynamic MAC Address Limits Per VLAN

Beginning with Cisco NX-OS Release 10.4(2)F, you can set the limit on number of dynamic MAC entries per VLAN to protect the control plane from MAC flood attacks on Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2/H2R/H1 platform switches.

Note The configuration is supported only on the default template and is not supported with the L2 heavy template.

SUMMARY STEPS

1. **config t**
2. **vlan** {*vlan-id* | *vlan-range*}
3. **mac address-table limit vlan** *vlan-id limit -value*
4. **exit**
5. **exit**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**vlan** {*vlan-id* | *vlan-range*}
	Example:
	`switch(config)# vlan 5`
	`switch(config-vlan)#`
		Places you into the VLAN configuration submode. If the VLAN does not exist, the system creates the specified VLAN and then enters the VLAN configuration submode.

Step 3	**mac address-table limit vlan** *vlan-id limit -value* **Example:** `switch(config-vlan)# mac address-table limit vlan 40 108`	Specifies the VLAN to which the MAC address limits should be applied. The permissible value for the limit is 100–196000. **Note** • This command is not supported on EoRs.

	Command or Action	Purpose
		• This command must not be used with vPC and/or VXLAN VLAN. • Enabling/Disabling the Mac-limit or modification of the mac-limit leads to flushing of all the dynamic MACs learned on that VLAN. However, the static or gateway MACs learning are not impacted. • The user is prompted to confirm before flushing.

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

Step 5	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits the configuration mode.

Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring L2 Heavy Mode

The purpose of this feature is to increase the current 92k MAC address scale to 200k by carving out a new L2-heavy template, changing FP tile hardware resource allocations, making necessary control plane changes and ISSU restore support to accommodate new scale.

Command	Purpose
sh system routing mode	Shows the configured and applied mode
system routing template-l2-heavy	Enables 200K MAC. 200K MAC is enabled only when this mode is configured and the system is reloaded. Use **no** form of this command to to disable this feature. **Note** Beginning with Cisco NX-OS Release 10.2(2)F, 200K MAC is supported on Cisco N9K-9332D-GX2B platform switches.
sh run | i system	Runs the applied mode

Guidelines & Limitations:

- This feature is supported for Layer 2 unidimensional scale only.
- SVI, Layer 3 interface, and VXLAN VLANs are not supported.
- Beginning with Cisco NX-OS Release 9.2(3), this feature is supported on the following platforms: N9K-C9264PQ, N9K-C9272Q, N9K-C9236C, N9K-C92300YC, N9K-C92304QC, N9K-C9232C, N9K-C92300YC and 9300-EX
- Beginning with Cisco NX-OS Release 10.2(2)F, the 200K MAC feature is supported on Cisco N9K-9332D-GX2B platform switches.
- Beginning with Cisco NX-OS Release 10.5(1)F, this feature is supported on Nexus N9K-9300-FX/FX2/FX3 TOR platforms.

Following is an example for configuring L2 heavy mode:

```
switch (config)# sh system routing mode
switch# Configured System Routing Mode: L2 Heavy
switch# Applied System Routing Mode: L2 Heavy
switch#
switch# show run | i system
switch# system routing template-l2-heavy
switch#
```

Verifying the Layer 2 Switching Configuration

To display Layer 2 switching configuration information, perform one of the following tasks:

Command	Purpose
show mac address-table	Displays information about the MAC address table.
show mac address-table limit	Displays information about the limits set for the MAC address table.
show mac address-table aging-time	Displays information about the aging time set for the MAC address entries. **Note** Beginning with Cisco NX-OS Release 10.2(1), Cisco Nexus 9000 and Nexus 3000 switches that use Cloudscale ASICs do not report the MAC age in show mac address outputs. The age column can be ignored as, instead of a fixed value of 0s that was reported in earlier releases, now, NA is reported. This is only a display limitation. MAC aging is still functionally enforced.

Command	Purpose
show mac address-table static	Displays information about the static entries on the MAC address table.
show mac address-table limit vlan	Displays information about the VLANs configured with MAC learn limit.
show interface [*interface*] **mac-address**	Displays the MAC addresses and the burn-in MAC address for the interfaces.
show forwarding consistency l2 {*module*}	Displays discrepant, missing, and extra MAC addresses between the tables on the module and the supervisor.

Configuration Example for Layer 2 Switching

The following example shows how to add a static MAC address and how to modify the default global aging time for MAC addresses:

```
switch# configure terminal
switch(config)# mac address-table static 0000.0000.1234 vlan 10 interface ethernet 2/15
switch(config)# mac address-table aging-time 120
```

The following example shows how to configure dynamic MAC limit per VLAN:

```
switch(config)# mac address-table limit vlan 251 100
Configuring MAC address limit will result in flushing existing Macs in the specified
VLAN/System. Proceed (yes/no)? [n] yes
Warning : MAC limit per VLAN feature isn't supported along with VPC/VxLAN. Please remove
the config if VPC/VxLAN config is present in this system !!!
switch(config)#
switch(config)# mac address-table limit vlan 252-253 100
Configuring MAC address limit will result in flushing existing Macs in the specified
VLAN/System. Proceed (yes/no)? [n] yes
switch(config)#
switch(config)# mac address-table limit vlan 254 300
Configuring MAC address limit will result in flushing existing Macs in the specified
VLAN/System. Proceed (yes/no)? [n] yes
```

Note When you configure this feature for the first time on the switch, a warning message stating that this feature is not supported with vPC/VXLAN is displayed. This warning message will not be displayed for the subsequent configurations.

To check the configured dynamic MAC limit and current count, use the following show command:

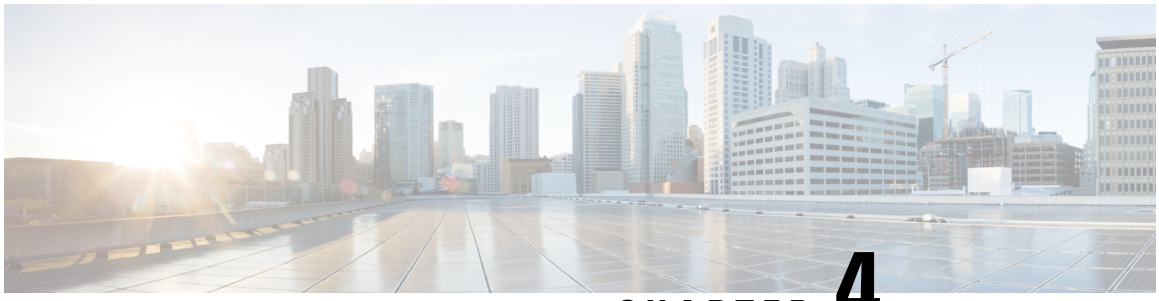
```
switch# show mac address-table limit vlan

Vlan       Conf Limit     Curr Count
----      ------------    ----------
```

```
251         100             100
252         100             100
253         100             75
254         300             60
```

Additional References for Layer 2 Switching -- CLI Version

Related Documents

Related Topic	Document Title
Static MAC addresses	*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*
Interfaces	*Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*
High availability	*Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide*
System management	*Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*



CHAPTER 4

Configuring Flex Links

This chapter describes how to configure Flex Links on the Cisco NX-OS 9000 Series Switches. Flex Links are a pair of interfaces that provide a mutual backup.

The chapter includes the following sections:

- Information About Flex Links, on page 27
- Guidelines and Limitations, on page 28
- Default Settings, on page 30
- Configuring Flex Links, on page 30
- Verifying Configuration, on page 34

Information About Flex Links

This section includes the following topics:

Flex Links

Flex Links are a pair of a Layer 2 interfaces (switchports or port channels), where one interface is configured to act as a backup to the other.

This feature provides an alternative solution to the Spanning Tree Protocol (STP), allowing users to turn off STP and still provide basic link redundancy. You generally configure Flex Links in networks where customers do not want to run STP on the switch. When you configure STP on the switch, it is not necessary to configure Flex Links because STP already provides link-level redundancy or backup.

Note STP is enabled by default on network node interfaces (NNIs). It is disabled on enhanced network interfaces (ENIs), but you can enable it. STP is not supported on user network interfaces (UNIs).

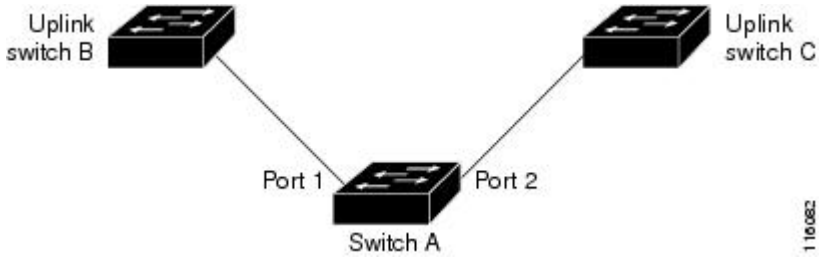
You configure Flex Links on one Layer 2 interface (the active link) by assigning another Layer 2 interface as the Flex Link or backup link. When one of the links is up and forwarding traffic, the other link is in standby mode, ready to begin forwarding traffic if the other link shuts down. At any given time, only one of the interfaces is in the linkup state and forwarding traffic. If the primary link shuts down, the standby link starts forwarding traffic. When the active link comes back up, it goes into standby mode and does not forward traffic. STP is disabled on Flex Link interfaces.

In Figure **Flex Links Configuration Example**, ports 1 and 2 on switch A are connected to uplink switches B and C. Because they are configured as Flex Links, only one of the interfaces is forwarding traffic; the other is in standby mode. If port 1 is the active link, it begins forwarding traffic between port 1 and switch B; the link between port 2 (the backup link) and switch C is not forwarding traffic. If port 1 goes down, port 2 comes up and starts forwarding traffic to switch C. When port 1 comes back up, it goes into standby mode and does not forward traffic; port 2 continues forwarding traffic.

Preemption

You can also choose to configure a preemption mechanism, specifying the preferred port for forwarding traffic. In the following figure, for example, you can configure the Flex Link pair with preemption mode so that after port 1 comes back up in the scenario, if it has greater bandwidth than port 2, port 1 begins forwarding after pre-empt delay (default pre-empt delay is 35 seconds); and port 2 becomes the standby. You do this by entering the interface configuration switchport backup interface preemption mode bandwidth and switchport backup interface preemption delay commands.

Figure 1: Flex Links Configuration Example



If a primary (forwarding) link or the standby link goes down, a trap notifies the network management stations. Flex Links are supported only on Layer 2 ports and port channels in either **trunk** or **access** mode. They are not supported on VLANs or Layer 3 ports.

Multicast

When a Flex Link interface is learned as an mrouter port, the standby (non-forwarding) interface is also co-learned as an mrouter port if the link is up. This co-learning is for internal software state maintenance and has no relevance with respect to IGMP operations or hardware forwarding unless multicast fast-convergence is enabled. With multicast fast-convergence configured, the co-learned mrouter port is immediately added to the hardware. Flex Link supports multicast fast convergence for IPv4 IGMP.

Guidelines and Limitations

Consider the following guidelines and limitations when configuring Flex Links:

- Flex links are supported on the following platforms : Cisco Nexus 9300-EX, 9300-FX , 9300-FX2 , C9364C switches.
- Flex Links are supported on Cisco Nexus 9300-FX, 9300-FX2, and 9348GC-FXP switches with IPv4 multicast.
- Because the Spanning Tree Protocol is implicitly disabled on Flex Link interfaces, ensure that you do not configure any other redundant paths in the same topology to prevent loops. In addition, configure

the corresponding links to upstream switches by using the spanning-tree port type normal command so they do not get blocked by Bridge Assurance.

- Flex Links are designed for uplink interfaces, which are typically configured as trunk ports. As a link backup mechanism, a Flex Link pair must have the same configuration characteristics, including the same switchport mode and list of allowed VLANs. Port-profile makes a convenient tool for syncing up such configurations for the Flex Link pair. Flex Link does not require that the two interfaces have the same configurations. However, long term mismatches in configurations may result in forwarding problems, particularly during failover.
- Flex Links cannot be configured on the following interface types:
  - Layer 3 interfaces
  - SPAN destinations
  - Port channel members
  - Interfaces configured with Private VLANs
  - Interfaces in end node mode
  - Layer 2 multi-path
- You can configure only one Flex Link backup link for any active link and it must be a different interface from the active interface.
- An interface can belong to only one Flex Link pair; it can be a backup link for only one active link.
- Neither of the links can be a port that belongs to an EtherChannel. However, you can configure two port channels (EtherChannel logical interfaces) as Flex Links, and you can configure a port channel and a physical interface as Flex Links, with either the port channel or the physical interface as the active link.
- A backup link does not have to be the same type (Ethernet or port channel) as the active link. However, we recommend that you configure both Flex Links with similar characteristics so that there are no loops or changes in behavior if the standby link begins to forward traffic.
- STP is disabled on Flex Link ports. A Flex Link port does not participate in STP, even if the VLANs present on the port are configured for STP. When STP is not enabled, be sure that there are no loops in the configured topology.

Note STP is available only on NNIs or ENIs.

- Do not configure any STP features (for example, PortFast, and BPDU Guard) on Flex Links ports.
- Default interface CLI on the flex link pair (active and standby) is not supported. When either breakout / in is performed in either primary or standby interface, flex link configuration is removed.
- vPC is not supported. Flex Link is used in place of vPC where configuration simplicity is desired and there is no need for active-active redundancy.
- Beginning with Cisco NX-OS Release 9.3(5), the Flex Link feature is supported on Cisco Nexus 9300-GX, N9K-C93108TC-FX3H, and N9K-C93108TC-FX3P platform switches.

- Beginning with Cisco NX-OS Release 9.3(7), the Flex Link feature is supported on Cisco N9K-C93180YC-FX3 platform switch.
- Beginning with Cisco NX-OS Release 10.1(1), the Flex Link feature is supported on Cisco N9K-C93180YC-FX3 platform switch.
- Beginning with Cisco NX-OS Release 10.2(2)F, the PVLAN and Flex Link features are supported on Cisco N9K-9332D-GX2B platform switch.
- Beginning with Cisco NX-OS Release 10.4(2)F, the Flex Link feature is supported on Cisco N9K-C93108TC-FX3 platform switch.

Default Settings

Parameters	Default
Flex links	Disabled
Multicast Fast-Convergence	Disabled
Flex links preemption mode	Off
Flex links preemption delay	35 seconds

Configuring Flex Links

Configuring Flexlinks

You can configure a pair of layer 2 interfaces (switch ports or port channels) as Flex Link interfaces, where one interface is configured to act as a backup to the other.

Before you begin

Review the Guidelines and Limitations for this feature. (See Guidelines and Limitations.)

SUMMARY STEPS

1. **configure terminal**
2. **feature flexlink**
3. **interface**{ **ethernet** *slot/ port* | **port-channel** *channel no*
4. **switchport backup interface** {**ethernet** *slot/ port* | **port-channel** *channel-no*} [**multicast fast-convergence**]
5. (Optional) **end**
6. (Optional) **show interface switchport backup**
7. (Optional) **copy running-config startup config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**feature flexlink**	Enables Flex Link.
Step 3	**interface**{ **ethernet** *slot/ port* | **port-channel** *channel no*	Specifies the Ethernet or port channel interface and enters interface configuration mode.
Step 4	**switchport backup interface** {**ethernet** *slot/ port* | **port-channel** *channel-no*} [**multicast fast-convergence**]	Specifies a physical layer 2 interface (Ethernet or port channel) as the backup interface in a Flex Link pair. When one link is forwarding traffic the other interface is in standby mode. • ethernet slot/port—Specifies the backup Ethernet interface. The slot number is 1 and the port number is from 1 to 48. • port-channel port-channel-no—Specifies the backup port channel interface. The port-channel-no number is from 1 to 4096. • multicast—Specifies the multicast parameters. • fast-convergence—Configures fast convergence on the backup interface.
Step 5	(Optional) **end**	Return to privileged EXEC mode.
Step 6	(Optional) **show interface switchport backup**	Verifies the configuration.
Step 7	(Optional) **copy running-config startup config**	Save your entries in the switch startup configuration file.

Example

This example shows how to configure an Ethernet switchport backup pair: Ethernet 1/1 is active interface, Ethernet 1/2 is the backup interface:

```
switch(config)# feature flexlink
switch(config)# interface ethernet 1/1
switch(config-if)# switchport backup interface ethernet 1/2
switch(config-if)# exit
switch(config)# interface port-channel300
switch(config-if)# switchport backup interface port-channel301
switch(config-if)# show ip igmp snooping mrouter
Type: S - Static, D - Dynamic, V - vPC Peer Link,
      I - Internal,C - Co-learned, U - User Configured
Vlan  Router-port   Type      Uptime      Expires
200   Po300          D         13:13:47    00:03:15
200   Po301          DC        13:13:47    00:03:15
```

This example shows how to configure the port channel switchport backup pair with multicast fast convergence:

```
switch(config)# interface port-channel10
switch(config-if)# switchport backup interface port-channel20 multicast fast-convergence
```

This example shows an example of multicast convergence with a pair of Flex Link interfaces: po305 and po306. A general query received on po305 makes it an mrouter port and po306 as co-learned.

```
switch(config)# interface po305
Switch(config-if)# switchport backup interface po306
switch# show ip igmp snooping mrouter
Type: S - Static, D - Dynamic, V - vPC Peer Link, I - Internal, C - Co-learned
Vlan  Router-port   Type      Uptime      Expires
4     Po300         D         00:00:12    00:04:50
4     Po301         DC        00:00:12    00:04:50
```

Configuring Flex Link Preemption

Configure a preemption scheme for the Flex Links pair (active and backup links).

Before you begin

Review the Guidelines and Limitations for this feature. (See Guidelines and Limitations.)

Define and enable the Flex Link. (See Configuring the Flex Link.)

Determine what preemption mode, if any, you want to assign to the port. (See Preemption.)

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot/port*
3. **switchport backup interface ethernet** *slot / port*
4. **switchport backup interface ethernet** *slot / port* **preemption mode {forced | bandwidth | off}**
5. **switchport backup interface ethernet** *slot / port* **preemption delay** *delay-time*
6. (Optional) **end**
7. (Optional) **show interface switchport backup**
8. (Optional) **copy running-config startup config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface ethernet** *slot/port*	Specify the interface, and enters interface configuration mode. The interface can be a physical Layer 2 interface or a port channel (logical interface).

	Command or Action	Purpose
Step 3	**switchport backup interface ethernet** *slot / port*	Configures a physical Layer 2 interface (or port channel) as part of a Flex Link pair with the interface. When one link is forwarding traffic, the other interface is in standby mode.
Step 4	**switchport backup interface ethernet** *slot / port* **preemption mode {forced | bandwidth | off}**	Configures a physical Layer 2 interface (Ethernet or port channel) as part of a flex link pair. When one link is forwarding traffic the other interface is in standby mode. • preemption—Configures a preemption scheme for a backup interface pair. • mode—Specifies the preemption mode. Configure a preemption mechanism and delay for a Flex Link pair. You can configure the preemption as: • **forced**—the active interface always preempts the backup. • **bandwidth**—the interface with the higher bandwidth always acts as the active interface. • **off**—no preemption happens from active to backup. **Note** During a bandwidth preemption mode, only bandwidth changes are considered, speed changes are ignored.
Step 5	**switchport backup interface ethernet** *slot / port* **preemption delay** *delay-time*	Configure the delay time until a port preempts another port. The delay-time range is from 1 to 300 seconds. The default preemption delay is 35 seconds. **Note** Setting a delay time only works with forced and bandwidth modes.
Step 6	(Optional) **end**	Return to privileged EXEC mode.
Step 7	(Optional) **show interface switchport backup**	Verifies the configuration.
Step 8	(Optional) **copy running-config startup config**	Save your entries in the switch startup configuration file.

Example

This example shows how to sets the preemption mode to forced, sets the delay time to 50, and verifies the configuration:

```
switch(config)# configure terminal
switch(config)# interface ethernet 1/48
switch(config-if)# switchport backup interface ethernet 1/4 preemption mode forced
switch(config-if)# switchport backup interface ethernet 1/4 preemption delay 50
switch(config-if)# end
switch# show interface switchport backup detail
```

```
Switch Backup Interface Pairs:

Active Interface        Backup Interface        State
------------------------------------------------------------------------

Ethernet1/48            Ethernet1/4             Active Down/Backup Down
        Preemption Mode  : forced
        Preemption Delay : 50 seconds
        Multicast Fast Convergence  : Off
        Bandwidth : 10000000 Kbit (Ethernet1/48), 10000000 Kbit (Ethernet1/4)
```

Verifying Configuration

Command	Purpose
show interface switchport backup	Displays information about all switchport Flex Link interfaces.
show interface switchport backup detail	Displays detailed information about all switchport Flex Link interfaces.
show running-config backup **show startup-config backup**	Displays the running or startup configuration for backup interfaces.
show running-config flexlink **show startup-config flexlink**	Displays the running or startup configuration for flex link interfaces.

This example shows summary configuration for the Flex Link pair:

```
9k-203-Pip(config)# show interface switchport backup


Switch Backup Interface Pairs:

Active Interface Backup Interface State
------------------------------------------------------------------------
Ethernet1/9 port-channel103 Active Standby/Backup Up
Ethernet1/12 Ethernet1/13 Active Up/Backup Standby
Ethernet1/21 port-channel203 Active Up/Backup Standby
Ethernet1/24 Ethernet1/25 Active Up/Backup Standby
port-channel301 port-channel302 Active Down/Backup Up


k-203-Pip(config)# show interface switchport backup detail


Switch Backup Interface Pairs:

Active Interface Backup Interface State
------------------------------------------------------------------------
Ethernet1/9 port-channel103 Active Standby/Backup Up
Preemption Mode : bandwidth
Preemption Delay : 1 seconds
Multicast Fast Convergence : On
Bandwidth : 1000000 Kbit (Ethernet1/9), 2000000 Kbit (port-channel103)
```

..

This example shows information about all switchport Flex Link interfaces:

```
switch# show interface switchport backup


Switch Backup Interface Pairs:

Active Interface        Backup Interface        State
------------------------------------------------------------------------
Ethernet1/1             Ethernet1/2             Active Down/Backup Down
Ethernet1/8             Ethernet1/45            Active Down/Backup Down
Ethernet1/48            Ethernet1/4             Active Down/Backup Down
port-channel10          port-channel20          Active Down/Backup Up
port-channel300         port-channel301         Active Down/Backup Down
```

This example shows details about all switchport Flex Link interfaces:

```
switch# show interface switchport backup detail


Switch Backup Interface Pairs:

Active Interface        Backup Interface        State
------------------------------------------------------------------------
Ethernet1/1             Ethernet1/2             Active Down/Backup Down
        Preemption Mode  : off
        Multicast Fast Convergence  : Off
        Bandwidth : 10000000 Kbit (Ethernet1/1), 10000000 Kbit (Ethernet1/2)

Ethernet1/8             Ethernet1/45            Active Down/Backup Down
        Preemption Mode  : forced
        Preemption Delay : 10 seconds
        Multicast Fast Convergence  : Off
        Bandwidth : 10000000 Kbit (Ethernet1/8), 10000000 Kbit (Ethernet1/45)

Ethernet1/48            Ethernet1/4             Active Down/Backup Down
        Preemption Mode  : forced
        Preemption Delay : 50 seconds
        Multicast Fast Convergence  : Off
        Bandwidth : 10000000 Kbit (Ethernet1/48), 10000000 Kbit (Ethernet1/4)

port-channel10          port-channel20          Active Down/Backup Up
        Preemption Mode  : forced
        Preemption Delay : 10 seconds
        Multicast Fast Convergence  : Off
        Bandwidth : 100000 Kbit (port-channel10), 10000000 Kbit (port-channel20)

port-channel300         port-channel301         Active Down/Backup Down
        Preemption Mode  : off
        Multicast Fast Convergence  : Off
        Bandwidth : 100000 Kbit (port-channel300), 100000 Kbit (port-channel301)
```

This example shows the running configuration for backup interfaces:

```
switch# show running-config backup

!Command: show running-config backup
!Time: Sun Mar  2 03:05:17 2014

version 6.0(2)A3(1)
feature flexlink
```

```
interface port-channel10
  switchport backup interface port-channel20 preemption mode forced
  switchport backup interface port-channel20 preemption delay 10

interface port-channel300
  switchport backup interface port-channel301

interface Ethernet1/1
  switchport backup interface Ethernet1/2

interface Ethernet1/8
  switchport backup interface Ethernet1/45 preemption mode forced
  switchport backup interface Ethernet1/45 preemption delay 10

interface Ethernet1/48
  switchport backup interface Ethernet1/4 preemption mode forced
  switchport backup interface Ethernet1/4 preemption delay 50
```

This example shows the startup configuration for backup interfaces:

```
switch# show startup-config backup

!Command: show startup-config backup
!Time: Sun Mar  2 03:05:35 2014
!Startup config saved at: Sun Mar  2 02:54:58 2014

version 6.0(2)A3(1)
feature flexlink


interface port-channel10
  switchport backup interface port-channel20 preemption mode forced
  switchport backup interface port-channel20 preemption delay 10

interface Ethernet1/8
  switchport backup interface Ethernet1/45 preemption mode forced
  switchport backup interface Ethernet1/45 preemption delay 10
```

This example shows the startup configuration for backup interfaces:

```
switch# show startup-config backup

!Command: show startup-config backup
!Time: Sun Mar  2 03:05:35 2014
!Startup config saved at: Sun Mar  2 02:54:58 2014

version 6.0(2)A3(1)
feature flexlink


interface port-channel10
  switchport backup interface port-channel20 preemption mode forced
  switchport backup interface port-channel20 preemption delay 10

interface Ethernet1/8
  switchport backup interface Ethernet1/45 preemption mode forced
  switchport backup interface Ethernet1/45 preemption delay 10
```

This example shows the running configuration of Flex Link:

```
switch# show running-config flexlink

!Command: show running-config flexlink
!Time: Sun Mar  2 03:11:49 2014

version 6.0(2)A3(1)
```

```
feature flexlink


interface port-channel10
  switchport backup interface port-channel20 preemption mode forced

interface port-channel300
  switchport backup interface port-channel301

interface port-channel305
  switchport backup interface port-channel306

interface Ethernet1/1
  switchport backup interface Ethernet1/2

interface Ethernet1/8
  switchport backup interface Ethernet1/45 preemption mode forced
  switchport backup interface Ethernet1/45 preemption delay 10

interface Ethernet1/48
  switchport backup interface Ethernet1/4 preemption mode forced
  switchport backup interface Ethernet1/4 preemption delay 50
```

This example shows the startup configuration of Flex Link:

```
switch# show startup-config flexlink

!Command: show startup-config flexlink
!Time: Sun Mar  2 03:06:00 2014
!Startup config saved at: Sun Mar  2 02:54:58 2014

version 6.0(2)A3(1)
feature flexlink


interface port-channel10
  switchport backup interface port-channel20 preemption mode forced
  switchport backup interface port-channel20 preemption delay 10

interface Ethernet1/8
  switchport backup interface Ethernet1/45 preemption mode forced
  switchport backup interface Ethernet1/45 preemption delay 10
```

Note Before using the **no feature flexlink**, all flexlink pair configuration must be disabled.

In order to ensure, user will be promoted with a confirmation message when user execute **no feature flexlink** as shown below:

```
"WARNING!!! Please remove all flexlink configuration before disabling feature flexlink.
Failure to do so may put ports in inconsistent state. Do you want to proceed? Y/N :"
```

This message is prompted only if DME is enabled in the system.

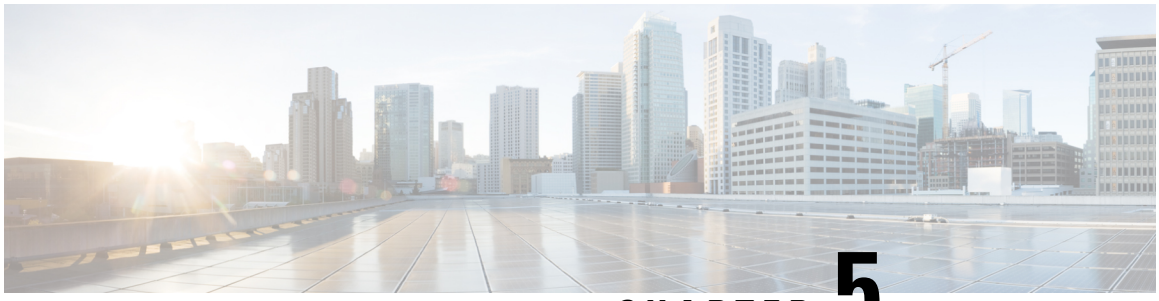
If the user chooses to proceed with this command, flexlink peer configuration will remain in the running configuration.

This, in turn, may cause system inconsistency in the ports, that are a part of flexlink configuration.

Once system is in an inconsistent state, the user needs to recover the system.

For recovery, the user needs to re-configure using the command **feature flexlink** and remove each interface pair configuration using the command **no switchport backup interface Ethernet x/y**.

Once all the pair configurations are removed, the user can execute **no feature flexlink**.



CHAPTER 5

Configuring VLANs

- Information About VLANs, on page 39
- Prerequisites for Configuring VLANs, on page 43
- Guidelines and Limitations for Configuring VLANs, on page 44
- Default Settings for VLANs, on page 45
- Configuring a VLAN, on page 45
- Verifying the VLAN Configuration, on page 54
- Displaying and Clearing VLAN Statistics, on page 55
- Configuration Example for VLANs, on page 55
- Additional References for VLANs, on page 55

Information About VLANs

You can use VLANs to divide the network into separate logical areas at the Layer 2 level. VLANs can also be considered as broadcast domains.

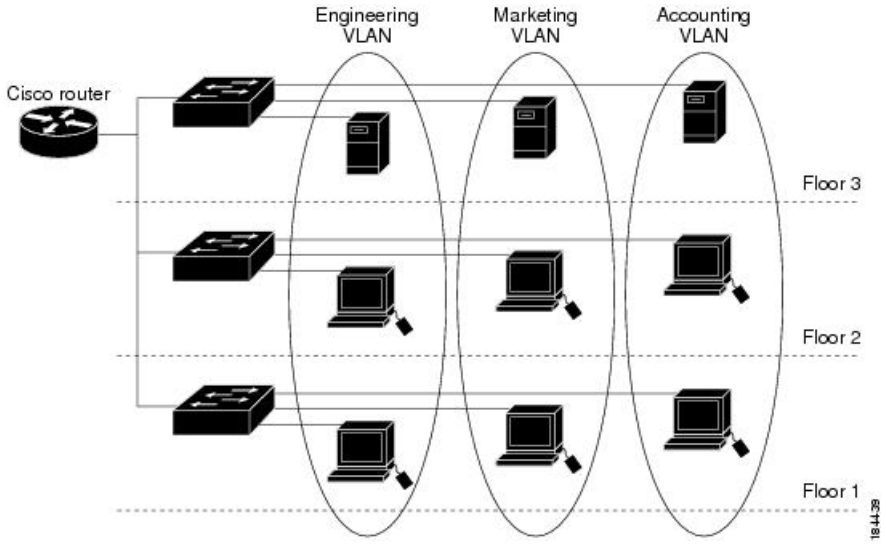
Any switch port can belong to a VLAN, and unicast broadcast and multicast packets are forwarded and flooded only to end stations in that VLAN. Each VLAN is considered a logical network, and packets destined for stations that do not belong to the VLAN must be forwarded through a router.

Understanding VLANs

A VLAN is a group of end stations in a switched network that is logically segmented by function or application, without regard to the physical locations of the users. VLANs have the same attributes as physical LANs, but you can group end stations even if they are not physically located on the same LAN segment.

Any switch port can belong to a VLAN, and unicast, broadcast, and multicast packets are forwarded and flooded only to end stations in that VLAN. Each VLAN is considered as a logical network, and packets destined for stations that do not belong to the VLAN must be forwarded through a router. The following figure shows VLANs as logical networks. The stations in the engineering department are assigned to one VLAN, the stations in the marketing department are assigned to another VLAN, and the stations in the accounting department are assigned to another VLAN.

Figure 2: VLANs as Logically Defined Networks



VLANs are usually associated with IP subnetworks. For example, all the end stations in a particular IP subnet belong to the same VLAN. To communicate between VLANs, you must route the traffic.

By default, a newly created VLAN is operational; that is, the newly created VLAN is in the no shutdown condition. Additionally, you can configure VLANs to be in the active state, which is passing traffic, or the suspended state, in which the VLANs are not passing packets. By default, the VLANs are in the active state and pass traffic.

A VLAN interface, or switched virtual interface (SVI), is a Layer 3 interface that is created to provide communication between VLANs. In order to route traffic between VLANs, you must create and configure a VLAN interface for each VLAN. Each VLAN requires only one VLAN interface.

VLAN Ranges

Note The extended system ID is always automatically enabled in Cisco Nexus 9000 devices.

The device supports up to 4095 VLANs in accordance with the IEEE 802.1Q standard. The software organizes these VLANs into ranges, and you use each range slightly differently.

For information about configuration limits, see the verified scalability limits documentation for your switch.

This table describes the VLAN ranges.

Table 3: VLAN Ranges

VLANs Numbers	Range	Usage
1	Normal	Cisco default. You can use this VLAN, but you cannot modify or delete it.
2—1005	Normal	You can create, use, modify, and delete these VLANs.

VLANs Numbers	Range	Usage
1006—3967	Extended	You can create, name, and use these VLANs. You cannot change the following parameters: • The state is always active. • The VLAN is always enabled. You cannot shut down these VLANs.
3968-4095	Internally allocated	These reserved VLANs are allocated for internal device use.

Note Cisco recommends that you enter the range in an increasing order, though the system accepts the range entered in decreasing order.

For example, to delete the range of VLANs from 1602 to 1607, the recommended way to enter the value is 1602-1607, rather than 1607-1602. Entering the range as 1602-7 will delete VLANs from 7 to 1602, instead of 1602 to 1607.

About Reserved VLANs

The following are notes about reserved VLANs (3968 to 4095):

- The software allocates a group of VLAN numbers for features like multicast and diagnostics, that need to use internal VLANs for their operation. By default, the system allocates a block of 128 reserved VLANs (3968 to 4095) for these internal uses.
- You can change the range of reserved VLANs with the system **vlan** *vlan-id* **reserve** command. This allows you to set a different range of VLANs to be used as the reserved VLANs. The selected VLANs must be reserved in groups of 128.
  - You may configure VLANs 3968-4092 for other purposes.
  - VLANs 4093-4095 are always reserved for internal use and cannot be used other purposes.

  For example,

  ```
  system vlan 400 reserve
  ```

  reserves VLANs 400-527.

  The new reserved range takes effect after the running configuration is saved and the device is reloaded.

  - VLANs 4093-4095 are always reserved for internal use and cannot be used other purposes.

    In the example, the result of the command would be that VLANs 400-527 are reserved and that VLANs 4093-4095 are also reserved.

- The **no system vlan** *vlan-id* **reserve** command changes the range for reserved VLANs to the default range of 3968-4095 after the device is reloaded.
- Use the **show system vlan reserved** command to verify the range of the current and future reserved VLAN ranges.

Example of VLAN Reserve

The following is an example of configuring the VLAN reserve (before and after image reload):

```
**********************************************
CONFIGURE NON-DEFAULT RANGE, "COPY R S" AND RELOAD
**********************************************
switch(config)# system vlan 400 reserve
"vlan configuration 400-527" will be deleted automatically.
Vlans, SVIs and sub-interface encaps for vlans 400-527 need to be removed by the user.
Continue anyway? (y/n) [no] y
Note: After switch reload, VLANs 400-527 will be reserved for internal use.
      This requires copy running-config to startup-config before
      switch reload.  Creating VLANs within this range is not allowed.

switch(config)# show system vlan reserved

system current running vlan reservation: 3968-4095

system future running vlan reservation: 400-527


switch(config)# copy running-config startup-config
[########################################] 100%

switch(config)# reload
This command will reboot the system. (y/n)?  [n] y

************
AFTER RELOAD
************

switch# show system vlan reserved

system current running vlan reservation: 400-527
```

Creating, Deleting, and Modifying VLANs

Note By default, all Cisco Nexus 9396 and Cisco Nexus 93128 ports are Layer 2 ports.

By default, all Cisco Nexus 9504 and Cisco Nexus 9508 ports are Layer 3 ports.

VLANs are numbered from 1 to 3967. All ports that you have configured as switch ports belong to the default VLAN when you first bring up the switch as a Layer 2 device. The default VLAN (VLAN1) uses only default values, and you cannot create, delete, or suspend activity in the default VLAN.

You create a VLAN by assigning a number to it; you can delete VLANs and move them from the active operational state to the suspended operational state. If you attempt to create a VLAN with an existing VLAN ID, the device goes into the VLAN submode but does not create the same VLAN again.

Newly created VLANs remain unused until Layer 2 ports are assigned to the specific VLAN. All the ports are assigned to VLAN1 by default.

Depending on the range of the VLAN, you can configure the following parameters for VLANs (except the default VLAN):

- VLAN name
- VLAN state
- Shutdown or not shutdown

You can configure VLAN long-names of up to 128 characters. To configure VLAN long-names, VTP must be in transparent mode.

Note See the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide* for information on configuring ports as VLAN access or trunk ports and assigning ports to VLANs.

When you delete a specified VLAN, the ports associated to that VLAN become inactive and no traffic flows. When you delete a specified VLAN from a trunk port, only that VLAN is shut down and traffic continues to flow on all the other VLANs through the trunk port.

However, the system retains all the VLAN-to-port mapping for that VLAN, and when you reenable or re-create, that specified VLAN, the system automatically reinstates all the original ports to that VLAN. The static MAC addresses and aging time for that VLAN are not restored when the VLAN is reenabled.

Note Commands entered in the VLAN configuration submode are not immediately executed. You must exit the VLAN configuration submode for configuration changes to take effect.

High Availability for VLANs

The software supports high availability for both stateful and stateless restarts, as during a cold reboot, for VLANs. For the stateful restarts, the software supports a maximum of three retries. If you try more than 3 times within 10 seconds of a restart, the software reloads the supervisor module.

You can upgrade or downgrade the software seamlessly when you use VLANs.

Note See the *Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide*, for complete information on high availability features.

Prerequisites for Configuring VLANs

VLANs have the following prerequisites:

- You must be logged onto the device.
- You must create the VLAN before you can do any modification of that VLAN.

Guidelines and Limitations for Configuring VLANs

VLANs have the following configuration guidelines and limitations:

- **show** commands with the **internal** keyword are not supported.
- You can configure a single VLAN or a range of VLANs.

  When you configure a large number of VLANs, first create the VLANs using the **vlan** command (for example, **vlan** *200-300*, *303-500*). After the VLANS have been successfully created, name or configure those VLANs sequentially.
- You cannot create, modify, or delete any VLANs that are within the group of VLANs reserved for internal use.
- VLAN1 is the default VLAN. You cannot create, modify, or delete this VLAN.
- VLANs 1006 to 3967 are always in the active state and are always enabled. You cannot suspend the state or shut down these VLANs.
- When the spanning tree mode is changed, the Layer 3 subinterface VLANs that share the same VLAN IDs with Layer 2 VLANs might be affected by a few micro-seconds of traffic drops as a result of the hardware re-programming.
- VLANs 3968 to 4095 are reserved for internal device use by default.
- A port *cannot* simultaneously have port-VLAN mapping and operate as a Private VLAN (PVLAN) interface in promiscuous, isolated, trunk, or host port modes due to hardware and configuration limitations. PVLAN and port-VLAN features operate independently on separate ports. You can configure and use the same VLAN for both functionalities. This is applicable on these releases.
  - Cisco NX-OS Release 10.2(9)M
  - Cisco NX-OS Release10.3(7)M
  - Cisco NX-OS Release 10.4(5)M
  - Cisco NX-OS Release 10.5(3)F
- Beginning with Cisco NX-OS Release 10.5(3)F, VLANs configuration is supported on the Cisco Nexus 93C64E-SG2-Q switch.
- Beginning with Cisco NX-OS Release 9.2(3), VLANs can be configured to have vn-segments.
- Cisco Nexus 9348GC-FX3PH switch has feature limitations due to half-duplex for ports 41-48.
- Cisco Nexus C93108TC-FX3 switch has feature limitations due to half-duplex for ports 41-48.
- QOS/ACL/SPAN are not supported on FEX HIFs.
- Beginning with Cisco NX-OS Release 9.3(9), PVLAN configuration is not allowed on vPC Peer-link interfaces.

Default Settings for VLANs

This table lists the default settings for VLAN parameters.

Table 4: Default VLAN Parameters

Parameters	Default
VLANs	Enabled
VLAN	VLAN1—A port is placed in VLAN1 when you configure it as a switch port.
VLAN ID	1
VLAN name	• Default VLAN (VLAN1)—default • All other VLANs—VLAN *vlan-id*
VLAN state	Active
STP	Enabled; Rapid PVST+ is enabled
VTP	Disabled
VTP version	1

Configuring a VLAN

Note See the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*, for information on assigning Layer 2 interfaces to VLANs (access or trunk ports). All interfaces are in VLAN1 by default.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Creating and Deleting a VLAN - CLI Version

You can create or delete all VLANs except the default VLAN and those VLANs that are internally allocated for use by the device.

Once a VLAN is created, it is automatically in the active state.

Note When you delete a VLAN, ports associated to that VLAN become inactive. Therefore, no traffic flows and the packets are dropped. On trunk ports, the port remains open and the traffic from all other VLANs except the deleted VLAN continues to flow.

If you create a range of VLANs and some of these VLANs cannot be created, the software returns a message listing the failed VLANs, and all the other VLANs in the specified range are created.

Note You can also create and delete VLANs in the VLAN configuration submode.

SUMMARY STEPS

1. **config t**
2. **vlan** {*vlan-id* | *vlan-range*}
3. **exit**
4. (Optional) **show vlan**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**vlan** {*vlan-id* | *vlan-range*}
	Example:
	`switch(config)# vlan 5`
	`switch(config-vlan)#`
		Creates a VLAN or a range or VLANs. If you enter a number that is already assigned to a VLAN, the device puts you into the VLAN configuration submode for that VLAN. If you enter a number that is assigned to an internally allocated VLAN, the system returns an error message. However, if you enter a range of VLANs and one or more of the specified VLANs is outside the range of internally allocated VLANs, the command takes effect on only those VLANs outside the range. The range is from 2 to 3967; VLAN1 is the default VLAN and cannot be created or deleted. You cannot create or delete those VLANs that are reserved for internal use. For more information about VLAN ranges, see VLAN Ranges, on page 40

Step 3	**exit** **Example:**	Exits the VLAN mode.

	Command or Action	Purpose
	`switch(config-vlan)# exit` `switch(config)#`	
Step 4	(Optional) **show vlan** **Example:** `switch# show vlan`	Displays information and status of VLANs.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to create a range of VLANs from 15 to 20:

```
switch# config t
switch(config)# vlan 15-20
switch(config-vlan)# exit
switch(config)#
```

Entering the VLAN Configuration Submode

To configure or modify the VLAN for the following parameters, you must be in the VLAN configuration submode:

- Name
- State
- Shut down

SUMMARY STEPS

1. **config t**
2. **vlan** {*vlan-id* | *vlan-range*}
3. **exit**
4. (Optional) **show vlan**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**config t** **Example:**	Enters configuration mode.

	Command or Action	Purpose
	`switch# config t` `switch(config)#`	

Step 2	**vlan** {*vlan-id* | *vlan-range*}
	Example:
	`switch(config)# vlan 5`
	`switch(config-vlan)#`
		Places you into the VLAN configuration submode. This submode allows you to name, set the state, disable, and shut down the VLAN or range of VLANs.
		You cannot change any of these values for VLAN1 or the internally allocated VLANs. For more information about VLAN ranges, see VLAN Ranges, on page 40

Step 3	**exit**
	Example:
	`switch(config-vlan)# exit`
	`switch(config)#`
		Exits the VLAN configuration mode.

Step 4	(Optional) **show vlan** **Example:** `switch# show vlan`	Displays information and status of VLANs.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to enter and exit the VLAN configuration submode:

```
switch# config t
switch(config)# vlan 15
switch(config-vlan)# exit
switch(config)#
```

Configuring a VLAN

To configure or modify a VLAN for the following parameters, you must be in the VLAN configuration submode:

- Name
- State
- Shut down

Note You cannot create, delete, or modify the default VLAN or the internally allocated VLANs. Additionally, some of these parameters cannot be modified on some VLANs.

SUMMARY STEPS

1. **config t**
2. **vlan** {*vlan-id* | *vlan-range*}
3. **name** *vlan-name*
4. **state** {**active** | **suspend**}
5. **no shutdown**
6. **exit**
7. (Optional) **show vlan**
8. (Optional) **show vtp status**
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**vlan** {*vlan-id* | *vlan-range*}
	Example:
	`switch(config)# vlan 5`
	`switch(config-vlan)#`
		Places you into the VLAN configuration submode. If the VLAN does not exist, the system creates the specified VLAN and then enters the VLAN configuration submode. For more information about VLAN ranges, see VLAN Ranges, on page 40

Step 3	**name** *vlan-name* **Example:** `switch(config-vlan)# name accounting`	Names the VLAN. You can enter up to 32 alphanumeric characters to name the VLAN. You cannot change the name of VLAN1 or the internally allocated VLANs. The default value is VLANxxxx where xxxx represent four numeric digits (including leading zeroes) equal to the VLAN ID number. **Note** 128-character names are supported (VLAN Long-Name).
Step 4	**state** {**active** | **suspend**} **Example:** `switch(config-vlan)# state active`	Sets the state of the VLAN to active or suspend. While the VLAN state is suspended, the ports associated with this VLAN become inactive, and that VLAN does not pass any traffic. The default state is active. You cannot suspend the state for the default VLAN or VLANs 1006 to 3967.
Step 5	**no shutdown** **Example:** `switch(config-vlan)# no shutdown`	Enables the VLAN. The default value is no shutdown (or enabled). You cannot shut down the default VLAN, VLAN1, or VLANs 1006 to 3967.

	Command or Action	Purpose

Step 6	**exit**
	Example:
	`switch(config-vlan)# exit`
	`switch(config)#`
		Exits the VLAN configuration submode.

Step 7	(Optional) **show vlan** **Example:** `switch# show vlan`	Displays information and status of VLANs.
Step 8	(Optional) **show vtp status** **Example:** `switch# show vtp status`	Displays information and status of VLAN Trunking Protocols (VTPs).
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration. **Note** Commands entered in the VLAN configuration submode are not immediately executed. You must exit the VLAN configuration submode for configuration changes to take effect.

Example

This example shows how to configure optional parameters for VLAN 5:

```
switch# config t
switch(config)# vlan 5
switch(config-vlan)# name accounting
switch(config-vlan)# state active
switch(config-vlan)# no shutdown
switch(config-vlan)# exit
switch(config)#
```

Configuring a VLAN Before Creating the VLAN

You can configure a VLAN before you create the VLAN. This procedure is used for IGMP snooping, VTP, and other configurations.

Note The **show vlan** command does not display these VLANs unless you create it using the **vlan** command.

SUMMARY STEPS

1. **config t**
2. **vlan configuration** {*vlan-id*}

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**vlan configuration** {*vlan-id*}
	Example:
	`switch(config)# vlan configuration 20`
	`switch(config-vlan-config)#`
		Allows you to configure VLANs without actually creating them.

Example

This example shows how to configure a VLAN before creating it:

```
switch# config t
switch(config)# vlan configuration 20
switch(config-vlan-config)#
```

Enabling the VLAN Long-Name

You can configure VLAN long-names of up to 128 characters.

Note When **system vlan long-name** is included in the start-up configuration, the Cisco Nexus 9000 Series switch boots up in VTP off mode.

To enable VTP transparent mode:

1. Disable VTP
2. Remove **system vlan long-name** from the start-up configuration
3. Re-enable VTP

Before you begin

VTP must be in transparent or in off mode. VTP cannot be in client or server mode. For more details about VTP, see Configuring VTP.

SUMMARY STEPS

1. **configure terminal**
2. **system vlan long-name**

3. (Optional) **copy running-config startup-config**
4. **show running-config vlan**

DETAILED STEPS

Procedure

	Command or Action	Purpose

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

Step 2	**system vlan long-name** **Example:** `switch(config)# system vlan long-name`	Allows you to enable VLAN names that have up to 128 characters. Use the **no** form of this command to disable this feature.
Step 3	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 4	**show running-config vlan** **Example:** `switch(config)# show running-config vlan`	Verifies that the system VLAN long-name feature is enabled.

Example

This example shows how to enable VLAN long-names.

```
switch# configure terminal
switch(config)# system vlan long-name
switch(config)# copy running config startup config
switch(config)# show running-config vlan
```

Configuring Inner VLAN and Outer VLAN Mapping on a Trunk Port

You can configure VLAN translation from an inner VLAN and an outer VLAN to a local (translated) VLAN on a port.

Notes for configuring inner VLAN and outer VLAN mapping:

- VLAN translation (mapping) is supported on Cisco Nexus 9000 Series switches with a Network Forwarding Engine (NFE). VLAN translation is supported on Cisco Nexus 9300-EX switches.
- Inner and outer VLAN cannot be on the trunk allowed list on a port where inner VLAN and outer VLAN is configured.

  For example:

```
switchport vlan mapping 11 inner 12 111
switchport trunk allowed vlan 11-12,111 /***Not valid because 11 is outer VLAN and 12
is inner VLAN.***/
```

- On the same port, no two mapping (translation) configurations can have the same outer (or original) or translated VLAN. Multiple inner VLAN and outer VLAN mapping configurations can have the same inner VLAN.

  For example:

```
switchport vlan mapping 101 inner 102 1001
switchport vlan mapping 101 inner 103 1002  /***Not valid because 101 is already used
as an original VLAN.***/
switchport vlan mapping 111 inner 104 1001  /***Not valid because 1001 is already used
 as a translated VLAN.***/
switchport vlan mapping 106 inner 102 1003  /***Valid because inner vlan can be the
same.***/
```

- Port VLAN mapping on a trunk port is supported on Cisco Nexus 9000 Series switches with a Network Forwarding Engine (NFE), Cisco Nexus 9200, 9300-EX, 9300-FX, and Cisco Nexus 9500 platform switches with EX/FX line cards.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type port*
3. [**no**] **switchport mode trunk**
4. **switchport vlan mapping enable**
5. **switchport vlan mapping** *outer-vlan-id* **inner** *inner-vlan-id translated-vlan-id*
6. (Optional) **copy running-config startup-config**
7. (Optional) **show interface** [*if-identifier*] **vlan mapping**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal**	Enters global configuration mode.
Step 2	**interface** *type port*	Enters interface configuration mode.
Step 3	[**no**] **switchport mode trunk**	Enters trunk configuration mode.
Step 4	**switchport vlan mapping enable**	Enables VLAN translation on the switch port. VLAN translation is disabled by default. **Note** Use the **no** form of this command to disable VLAN translation.

	Command or Action	Purpose
Step 5	**switchport vlan mapping** *outer-vlan-id* **inner** *inner-vlan-id translated-vlan-id*	Translates inner VLAN and outer VLAN to another VLAN.
Step 6	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration. **Note** The VLAN translation configuration does not become effective until the switch port becomes an operational trunk port
Step 7	(Optional) **show interface** [*if-identifier*] **vlan mapping**	Displays VLAN mapping information for a range of interfaces or for a specific interface.

Example

This example shows how to configure translation of double tag VLAN traffic (inner VLAN 12; outer VLAN 11) to VLAN 111.

```
switch# config t
switch(config)# interface ethernet1/1
switch(config-if)# switchport mode trunk
switch(config-if)# switchport vlan mapping enable
switch(config-if)# switchport vlan mapping 11 inner 12 111
switch(config-if)# switchport trunk allowed vlan 101-170
switch(config-if)# no shutdown

switch(config-if)# show mac address-table dynamic vlan 111

Legend:
        * - primary entry, G - Gateway MAC, (R) - Routed MAC, O - Overlay MAC
        age - seconds since last seen,+ - primary entry using vPC Peer-Link,
        (T) - True, (F) - False
   VLAN     MAC Address      Type      age     Secure NTFY Ports
---------+-----------------+--------+---------+------+----+------------------
*  111     0000.0092.0001   dynamic  0         F      F    nve1(100.100.100.254)
*  111     0000.0940.0001   dynamic  0         F      F    Eth1/1
```

Verifying the VLAN Configuration

To display VLAN configuration information, perform one of the following tasks:

Command	Purpose
show running-config vlan *vlan-id*	Displays VLAN information.
show vlan [**all-ports** | **brief** | **id** *vlan-id* | **name** *name* | **dot1q tag native**]	Displays VLAN information.
show vlan summary	Displays a summary of VLAN information.

Command	Purpose
show vtp status	Displays VTP information.

Displaying and Clearing VLAN Statistics

To display VLAN configuration information, perform one of the following tasks:

Command	Purpose
clear vlan [**id** *vlan-id*] **counters**	Clears counters for all VLANs or for a specified VLAN.
show vlan counters	Displays information on Layer 2 packets in each VLAN.

Configuration Example for VLANs

The following example shows how to create and name a VLAN as well as how to make the state active and administratively up:

```
switch# configure terminal
switch(config)# vlan 10
switch(config-vlan)# name test
switch(config-vlan)# state active
switch(config-vlan)# no shutdown
switch(config-vlan)# exit
switch(config)#
```

Additional References for VLANs

Related Documents

Related Topic	Document Title
NX-OS Layer 2 switching configuration	*Cisco Nexus 9000 Series NX-OS Layer 2 Switching Configuration Guide*
Interfaces, VLAN interfaces, IP addressing, and port channels	*Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*
Multicast routing	*Cisco Nexus 9000 Series NX-OS Multicast Routing Configuration Guide*
NX-OS fundamentals	*Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide*

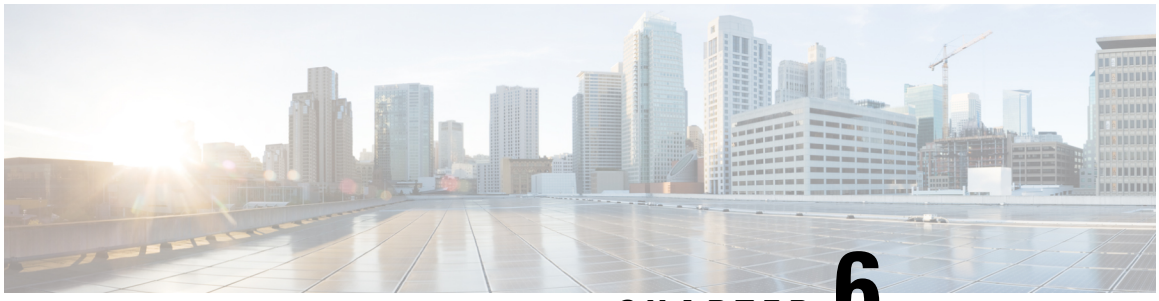
Related Topic	Document Title
High availability	*Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide*
System management	*Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*

Standards

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

MIBs

MIBs	MIBs Link
CISCO-VLAN-MEMBERSHIP MIB: • vmMembership Table • MIBvmMembershipSummaryTable • MIBvmMembershipSummaryTable	To locate and download MIBs, go to the following URL: https://cisco.github.i supportlists/nexus9000/Nexus9000MIBSupportList.html



CHAPTER 6

Configuring VTP

- Information About VTP, on page 57
- Guidelines and Limitations for Configuring VTP, on page 58
- Default Settings, on page 59
- Configuring VTP, on page 59

Information About VTP

VTP is supported for VTP version 1 and 2.

Note You can configure VLANs without actually creating the VLANs. For more details, see Configuring a VLAN Before Creating the VLAN, on page 50.

VTP

VTP is a Layer 2 messaging protocol that maintains VLAN consistency by managing the addition, deletion, and renaming of VLANs within a VTP domain. A VTP domain is made up of one or more network devices that share the same VTP domain name and that are connected with trunk interfaces. Each network device can be in only one VTP domain.

Layer 2 trunk interfaces, Layer 2 port channels, and virtual port channels (vPCs) support VTP functionality.

The VTP is disabled by default on the device. You can enable and configure VTP using the command-line interface (CLI). When VTP is disabled, the device does not relay any VTP protocol packets.

Note VTP worked only in transparent mode in the Cisco Nexus 9000 Series devices, allowing you to extend a VTP domain across the device.

When the device is in the VTP transparent mode, the device relays all VTP protocol packets that it receives on a trunk port to all other trunk ports. When you create or modify a VLAN that is in VTP transparent mode, those VLAN changes affect only the local device. A VTP transparent network device does not advertise its VLAN configuration and does not synchronize its VLAN configuration based on received advertisements.

Note VLAN 1 is required on all trunk ports used for switch interconnects if VTP is supported in the network. Disabling VLAN 1 from any of these ports prevents VTP from functioning properly.

VTP Overview

VTP allows each router or LAN device to transmit advertisements in frames on its trunk ports. These frames are sent to a multicast address where they can be received by all neighboring devices. They are not forwarded by normal bridging procedures. An advertisement lists the sending device's VTP management domain, its configuration revision number, the VLANs which it knows about, and certain parameters for each known VLAN. By hearing these advertisements, all devices in the same management domain learn about any new VLANs that are configured in the transmitting device. This process allows you to create and configure a new VLAN only on one device in the management domain, and then that information is automatically learned by all the other devices in the same management domain.

Once a device learns about a VLAN, the device receives all frames on that VLAN from any trunk port by default, and if appropriate, forwards them to each of its other trunk ports, if any. This process prevents unnecessary VLAN traffic from being sent to a device.

VTP also publishes information about the domain and the mode in a shared local database that can be read by other processes such as Cisco Discovery Protocol (CDP).

VTP Modes

VTP is supported in these modes:

- Transparent—Allows you to relay all VTP protocol packets that it receives on a trunk port to all other trunk ports. When you create or modify a VLAN that is in VTP transparent mode, those VLAN changes affect only the local device. A VTP transparent network device does not advertise its VLAN configuration and does not synchronize its VLAN configuration based on received advertisements.

If VTP is in transparent mode, you can configure VLAN long names of up to 128 characters.

VTP Per Interface

VTP allows you to enable or disable the VTP protocol on a per-port basis to control the VTP traffic. When a trunk is connected to a switch or end device, it drops incoming VTP packets and prevents VTP advertisements on this particular trunk. By default, VTP is enabled on all the switch ports.

Guidelines and Limitations for Configuring VTP

VTP has the following configuration guidelines and limitations:

- **show** commands with the **internal** keyword are not supported.
- In SNMP, the vlanTrunkPortVtpEnabled object indicates whether the VTP feature is enabled or not. The status of the vlanTrunkPortVtpEnabled object aligns with the output of the **show vtp trunk interface eth a/b** command.

- VTP advertisements are not sent out on Cisco Nexus Fabric Extender ports.
- VTP pruning is not possible with transparent devices. When there are transparent devices in a VTP domain, VTP pruning has to be disabled. If VTP pruning is not disabled on the neighboring devices, the Cisco Nexus devices will not learn any MACs from the neighboring device because the VLANs are pruned/disabled on the links pointing to the Nexus.

Default Settings

This table lists the default settings for VTP parameters.

Table 5: Default VTP Parameters

Parameters	Default
VTP	Disabled
VTP Mode	Transparent
VTP Domain	blank
VTP Version	1
VTP per Interface	Enabled

Configuring VTP

You can configure VTP on Cisco Nexus 9000 devices.

Note VLAN 1 is required on all trunk ports used for switch interconnects if VTP is used in transparent mode in the network. Disabling VLAN 1 from any of these ports prevents VTP from functioning properly in transparent mode.

Note VTP worked only in transparent mode.

SUMMARY STEPS

1. **config t**
2. **feature vtp**
3. **vtp domain** *domain-name*
4. **vtp version** {**1** | **2**}
5. **vtp file** *file-name*
6. **vtp password** *password-value*

7. **exit**
8. (Optional) **show vtp status**
9. (Optional) **show vtp counters**
10. (Optional) **show vtp interface**
11. (Optional) **show vtp password**
12. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**feature vtp**
	Example:
	`switch(config)# feature vtp`
	`switch(config)#`
		Enables VTP on the device. The default is disabled.

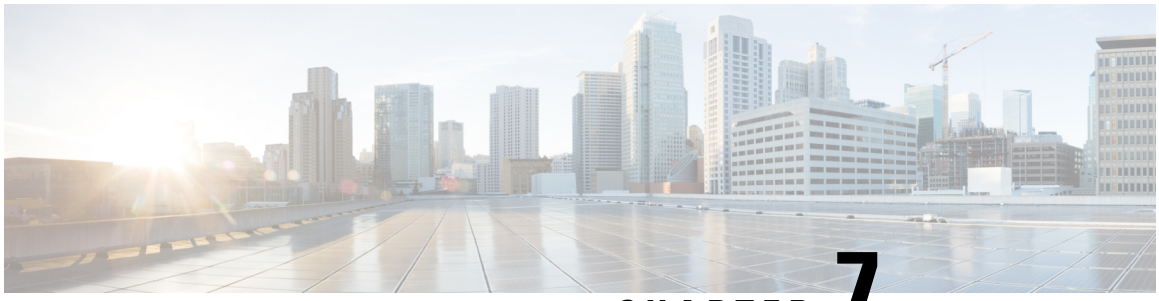
Step 3	**vtp domain** *domain-name* **Example:** `switch(config)# vtp domain accounting`	Specifies the name of the VTP domain that you want this device to join. The default is blank.
Step 4	**vtp version** {**1** | **2**} **Example:** `switch(config)# vtp version 2`	Sets the VTP version that you want to use. The default is version 1.
Step 5	**vtp file** *file-name* **Example:** `switch(config)# vtp file vtp.dat`	Specifies the ASCII filename of the IFS file system file where the VTP configuration is stored.
Step 6	**vtp password** *password-value* **Example:** `switch(config)# vtp password cisco`	Specifies the password for the VTP administrative domain.

Step 7	**exit**
	Example:
	`switch(config)# exit`
	`switch#`
		Exits the configuration submode.

Step 8	(Optional) **show vtp status** **Example:** `switch# show vtp status`	Displays information about the VTP configuration on the device, such as the version, mode, and revision number.

	Command or Action	Purpose
Step 9	(Optional) **show vtp counters** **Example:** `switch# show vtp counters`	Displays information about VTP advertisement statistics on the device.
Step 10	(Optional) **show vtp interface** **Example:** `switch# show vtp interface`	Displays the list of VTP-enabled interfaces.
Step 11	(Optional) **show vtp password** **Example:** `switch# show vtp password`	Displays the password for the management VTP domain.
Step 12	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Configuring VTP



CHAPTER 7

Configuring Private VLANs Using NX-OS

- Information About Private VLANs, on page 63
- Prerequisites for Private VLANs, on page 71
- Guidelines and Limitations for Configuring Private VLANs, on page 71
- Default Settings for Private VLANs, on page 74
- Configuring a Private VLAN, on page 74
- Verifying the Private VLAN Configuration, on page 90
- Displaying and Clearing Private VLAN Statistics, on page 90
- Configuration Examples for Private VLANs, on page 90
- Additional References for Private VLANs -- CLI Version, on page 91

Information About Private VLANs

Note You must enable the private VLAN feature before you can configure this feature.

Note A Layer 2 port can function as either a trunk port, an access port, or a private VLAN port.

In certain instances where similar systems do not need to interact directly, private VLANs provide additional protection at the Layer 2 level. Private VLANs are an association of primary and secondary VLANs.

A primary VLAN defines the broadcast domain with which the secondary VLANs are associated. The secondary VLANs may either be isolated VLANs or community VLANs. Hosts on isolated VLANs communicate only with associated promiscuous ports in primary VLANs, and hosts on community VLANs communicate only among themselves and with associated promiscuous ports but not with isolated ports or ports in other community VLANs.

In configurations that use integrated switching and routing functions, you can assign a single Layer 3 VLAN network interface to each private VLAN to provide routing. The VLAN network interface is created for the primary VLAN. In such configurations, all secondary VLANs communicate at Layer 3 only through a mapping with the VLAN network interface on the primary VLAN. Any VLAN network interfaces previously created on the secondary VLANs are put out-of-service.

Private VLAN Overview

You must enable private VLANs before the device can apply the private VLAN functionality.

You cannot disable private VLANs if the device has any operational ports in a private VLAN mode.

Note You must have already created the VLAN before you can convert the specified VLAN to a private VLAN, either primary or secondary.

Primary and Secondary VLANs in Private VLANs

The private VLAN feature addresses two problems that users encounter when using VLANs:

- Each VDC supports up to 4096 VLANs. If a user assigns one VLAN per customer, the number of customers that the service provider can support is limited.
- To enable IP routing, each VLAN is assigned with a subnet address space or a block of addresses, which can result in wasting the unused IP addresses and creating IP address management problems.

Using private VLANs solves the scalability problem and provides IP address management benefits and Layer 2 security for customers.

The private VLAN feature allows you to partition the Layer 2 broadcast domain of a VLAN into subdomains. A subdomain is represented by a pair of private VLANs: a primary VLAN and a secondary VLAN. A private VLAN domain can have multiple private VLAN pairs, one pair for each subdomain. All VLAN pairs in a private VLAN domain share the same primary VLAN. The secondary VLAN ID differentiates one subdomain from another.

Note A private VLAN domain has only one primary VLAN.

Secondary VLANs provide Layer 2 isolation between ports within the same private VLAN. The following two types are secondary VLANs within a primary VLAN:

- Isolated VLANs—Ports within an isolated VLAN cannot communicate with each other at the Layer 2 level.
- Community VLANs—Ports within a community VLAN can communicate with each other but cannot communicate with ports in other community VLANs or in any isolated VLANs at the Layer 2 level.

Private VLAN Ports

Note Both community and isolated private VLAN ports are labeled as PVLAN host ports. A PVLAN host port is either a community PVLAN port or an isolated PVLAN port depending on the type of secondary VLAN with which it is associated.

The types of private VLAN ports are as follows:

- Promiscuous port—A promiscuous port belongs to the primary VLAN. The promiscuous port can communicate with all interfaces, including the community and isolated host ports, that belong to those secondary VLANs associated to the promiscuous port and associated with the primary VLAN. You can have several promiscuous ports in a primary VLAN. Each promiscuous port can have several secondary VLANs, or no secondary VLANs, associated to that port. You can associate a secondary VLAN to more than one promiscuous port, as long as the promiscuous port and secondary VLANs are within the same primary VLAN. You may want to do this association for load balancing or redundancy purposes. You can also have secondary VLANs that are not associated to any promiscuous port, but these secondary VLANs cannot communicate to the Layer 3 interface.

Note As a best practice, you should map all the secondary ports on the primary to minimize any loss of traffic.

- Promiscuous trunk—You can configure a promiscuous trunk port to carry traffic for multiple primary VLANs. You map the private VLAN primary VLAN and either all or selected associated VLANs to the promiscuous trunk port. Each primary VLAN and one associated secondary VLAN is a private VLAN pair, For maximum PVLAN mappings, see *Verified Scalability Guide*.

Note Private VLAN promiscuous trunk ports carry traffic for normal VLANs as well as for primary private VLANs.

- Isolated port—An isolated port is a host port that belongs to an isolated secondary VLAN. This port has complete Layer 2 isolation from other ports within the same private VLAN domain, except that it can communicate with associated promiscuous ports. Private VLANs block all traffic to isolated ports except traffic from promiscuous ports. Traffic received from an isolated port is forwarded only to promiscuous ports. You can have more than one isolated port in a specified isolated VLAN, and each port is completely isolated from all other ports in the isolated VLAN.
- Isolated or secondary trunk—You can configure an isolated trunk port to carry traffic for multiple isolated VLANs. Each secondary VLAN on an isolated trunk port must be associated with a different primary VLAN. You cannot put two secondary VLANs that are associated with the same primary VLAN on an isolated trunk port. Each primary VLAN and one associated secondary VLAN is a private VLAN pair. For maximum PVLAN associations, see *Verified Scalability Guide*.

Note Private VLAN isolated trunk ports carry traffic for normal VLANs as well as for secondary private VLANs.

- Community port—A community port is a host port that belongs to a community secondary VLAN. Community ports communicate with other ports in the same community VLAN and with associated promiscuous ports. These interfaces are isolated at Layer 2 from all other interfaces in other communities and from all isolated ports within the private VLAN domain.

Note Because trunks can support the VLANs that carry traffic between promiscuous, isolated, and community ports, the isolated and community port traffic might enter or leave the device through a trunk interface.

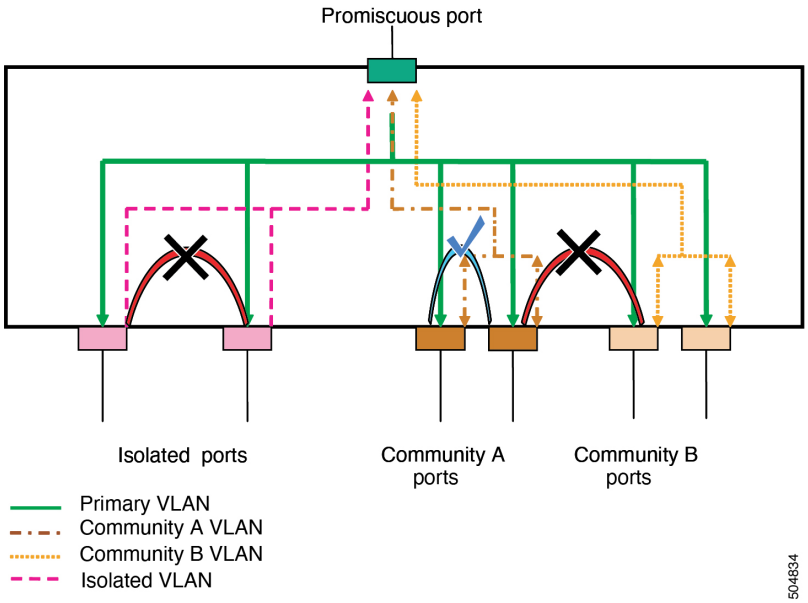
Primary, Isolated, and Community Private VLANs

Because the primary VLAN has the Layer 3 gateway, you associate secondary VLANs with the primary VLAN in order to communicate outside the private VLAN. Primary VLANs and the two types of secondary VLANs, isolated VLANs and community VLANs, have these characteristics:

- Primary VLAN— The primary VLAN carries traffic from the promiscuous ports to the (isolated and community) host ports and to other promiscuous ports.
- Isolated VLAN —An isolated VLAN is a secondary VLAN that carries unidirectional traffic upstream from the hosts toward the promiscuous ports and the Layer 3 gateway. You can configure one isolated VLAN in a primary VLAN. In addition, each isolated VLAN can have several isolated ports, and the traffic from each isolated port also remains completely separate.
- Community VLAN—A community VLAN is a secondary VLAN that carries upstream traffic from the community ports to the promiscuous port gateways and to other host ports in the same community. You can configure multiple community VLANs in a private VLAN domain. The ports within one community can communicate, but these ports cannot communicate with ports in any other community or isolated VLAN in the private VLAN.

Figure 3: Private VLAN Layer 2 Traffic Flows

This figure shows the Layer 2 traffic flows within a primary, or private VLAN, along with the types of VLANs and types of ports.



Note The private VLAN traffic flows are unidirectional from the host ports to the promiscuous ports. Traffic that egresses the promiscuous port acts like the traffic in a normal VLAN, and there is no traffic separation among the associated secondary VLAN.

A promiscuous port can serve only one primary VLAN, one isolated VLAN, and multiple community VLANs. (Layer 3 gateways are connected to the device through a promiscuous port.) With a promiscuous port, you can connect a wide range of devices as access points to a private VLAN. For example, you can use a promiscuous port to monitor or back up all the private VLAN servers from an administration workstation.

Note You can configure private VLAN promiscuous and isolated trunk ports. These promiscuous and isolated trunk ports carry traffic for multiple primary and secondary VLANs as well as normal VLAN.

Although you can have several promiscuous ports in a primary VLAN, you can have only one Layer 3 gateway per primary VLAN.

In a switched environment, you can assign an individual private VLAN and associated IP subnet to each individual or common group of end stations. The end stations need to communicate only with a default gateway to communicate outside the private VLAN.

Note You must enable the VLAN interface feature before you can configure the Layer 3 gateway. See the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide* for complete information on VLAN network interfaces and IP addressing.

Associating Primary and Secondary VLANs

To allow the host ports in secondary VLANs to communicate outside the private VLAN, you associate secondary VLANs to the primary VLAN. If the association is not operational, the host ports (isolated and community ports) in the secondary VLAN are brought down.

Note You can associate a secondary VLAN with only one primary VLAN.

For an association to be operational, the following conditions must be met:

- The primary VLAN must exist.
- The secondary VLAN must exist.
- The primary VLAN must be configured as a primary VLAN.
- The secondary VLAN must be configured as either an isolated or community VLAN.

Note See the **show** command display to verify that the association is operational. The device does not issue an error message when the association is nonoperational.

If you delete either the primary or secondary VLAN, the ports that are associated with the VLAN become inactive. When you reconvert the specified VLAN to private VLAN mode, the original associations are reinstated.

If the association is not operational on private VLAN trunk ports, only that VLAN goes down, not the entire port.

When you enter the **no private-vlan** command, the VLAN returns to the normal VLAN mode. All associations on that VLAN are suspended, but the interfaces remain in private VLAN mode.

If you enter the **no vlan** command for the primary VLAN, all private VLAN associations with that VLAN are lost. However, if you enter the **no vlan** command for a secondary VLAN, the private VLAN associations with that VLAN are suspended and return when you recreate the specified VLAN and configure it as the secondary VLAN.

Note This behavior is different from how Catalyst devices work.

In order to change the association between a secondary and primary VLAN, you must first remove the current association and then add the desired association.

Broadcast Traffic in Private VLANs

Broadcast traffic from ports in a private VLAN flows in the following ways:

- The broadcast traffic flows from all promiscuous ports to all ports in the primary VLAN. This broadcast traffic is distributed to all ports within the primary VLAN, including those ports that are not configured with private VLAN parameters.
- The broadcast traffic from all isolated ports is distributed only to those promiscuous ports in the primary VLAN that are associated to that isolated port.
- The broadcast traffic from community ports is distributed to all ports within the port's community and to all promiscuous ports that are associated to the community port. The broadcast packets are not distributed to any other communities within the primary VLAN or to any isolated ports.

Private VLAN Port Isolation

You can use private VLANs to control access to end stations as follows:

- Configure selected interfaces connected to end stations as isolated ports to prevent any communication at Layer 2. For example, if the end stations are servers, this configuration prevents Layer 2 communication between the servers.
- Configure interfaces connected to default gateways and selected end stations (for example, backup servers) as promiscuous ports to allow all end stations access to a default gateway.

Private VLANs and VLAN Interfaces

A VLAN interface to a Layer 2 VLAN is also called a switched virtual interface (SVI). Layer 3 devices communicate with a private VLAN only through the primary VLAN and not through secondary VLANs.

Configure VLAN network interfaces only for primary VLANs. Do not configure VLAN interfaces for secondary VLANs. VLAN network interfaces for secondary VLANs are inactive while the VLAN is configured as a secondary VLAN. You will see the following actions if you misconfigure the VLAN interfaces:

- If you try to configure a VLAN with an active VLAN network interface as a secondary VLAN, the configuration is not allowed until you disable the VLAN interface.
- If you try to create and enable a VLAN network interface on a VLAN that is configured as a secondary VLAN, that VLAN interface remains disabled and the system returns an error.

When the primary VLAN is associated with and mapped to the secondary VLAN, any configuration on the primary VLAN is propagated to the secondary VLANs. For example, if you assign an IP subnet to the VLAN network interface on the primary VLAN, this subnet is the IP subnet address of the entire private VLAN.

Note You must enable the VLAN interface feature before you configure VLAN interfaces. See the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*, for information on VLAN interfaces and IP addressing.

Private VLANs Across Multiple Devices

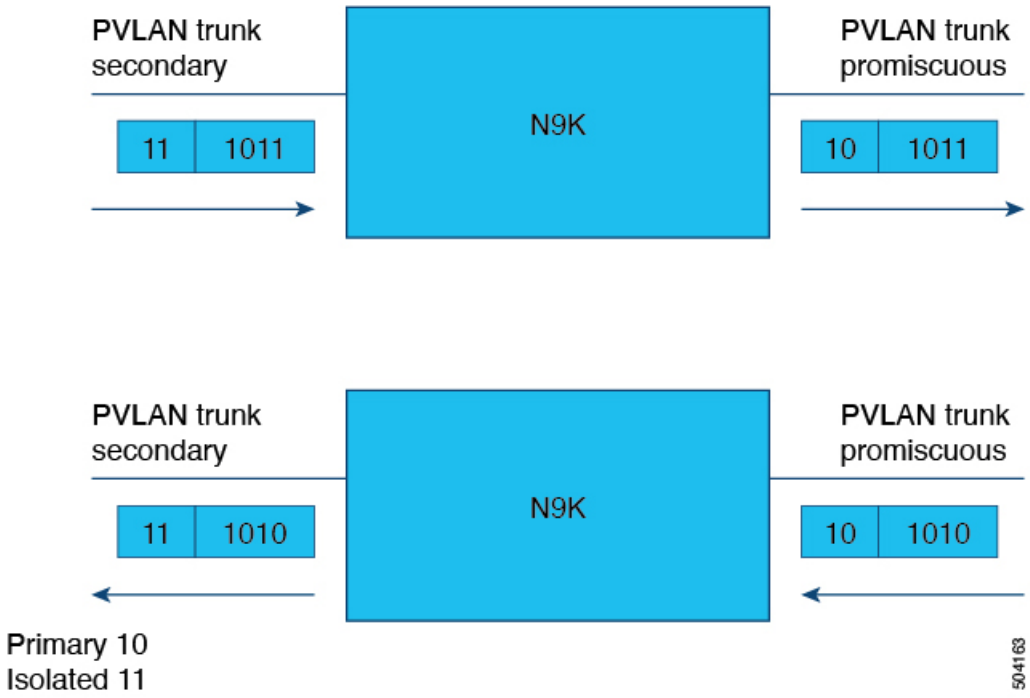
You can extend private VLANs across multiple devices by trunking the primary, isolated, and community VLANs to other devices that support private VLANs. To maintain the security of your private VLAN configuration and to avoid other uses of the VLANs configured to be private VLANs, configure private VLANs on all intermediate devices, including devices that have no private VLAN ports.

Private VLAN with Inner VLAN Tag Preservation

Beginning with Cisco NX-OS Release 10.2(3)F, if you have configured the global **system dot1q-tunnel transit** <*vlan*> command on a supported Cisco Nexus switch that acts as a transit box, then the packets coming in on private vlan trunk ports with 2 or more tags are preserved and sent out without stripping any of the inner tags. For more information about the command, refer to *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide* of the relevant release on cisco.com.

Note Inner tag preservation does not work when PVLAN and QinQ are configured on the same port.

The following figure illustrates the inner tag preservation on the supported Cisco Nexus switch when the packet moves from PVLAN secondary trunk to PVLAN promiscuous trunk and back.



A sample configuration is as follows:

```
vlan 10
private-vlan primary
private-vlan association 11-12
vlan 11
private-vlan isolated
vlan 12
private-vlan community

interface Ethernet1/1
switchport
switchport mode private-vlan trunk secondary
switchport private-vlan association trunk 10 11
no shutdown

interface Ethernet1/2
switchport
switchport mode private-vlan trunk promiscuous
switchport private-vlan mapping trunk 10 11-12
no shutdown

(config)# system dot1q-tunnel transit vlan 10,11
```

High Availability for Private VLANs

The software supports high availability for both stateful and stateless restarts, as during a cold reboot, for private VLANs. For the stateful restarts, the software supports a maximum of three retries. If you try more than 3 times within 10 seconds of a restart, the software reloads the supervisor module.

Note See the *Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide*, for complete information on high-availability features.

Prerequisites for Private VLANs

Private VLANs have the following prerequisites:

- You must be logged onto the device.
- You must enable the private VLAN feature.

Guidelines and Limitations for Configuring Private VLANs

Private VLANs (PVLANs) have the following configuration guidelines and limitations:

- When changing PVLAN host-association from one association to another, an error is displayed when the port-security static MAC address are not deleted. We recommend you delete the port-security static MAC address.

  ```
  ERROR: Static Port-Security Mac configured. Remove configured static port-security mac
   under the interfaces before changing the private-vlan
  ```

- When changing the PVLAN mapping on the vPC Port-channel in the promiscuous mode, the vPC PO member on vPC secondary flaps.
- **show** commands with the **internal** keyword are not supported.
- You must enable PVLANs before the device can apply the PVLAN functionality.
- A port *cannot* simultaneously have port-VLAN mapping and operate as a Private VLAN (PVLAN) interface in promiscuous, isolated, trunk, or host port modes due to hardware and configuration limitations. PVLAN and port-VLAN features operate independently on separate ports. You can configure and use the same VLAN for both functionalities. This is applicable on these releases.
  - Cisco NX-OS Release 10.2(9)M
  - Cisco NX-OS Release10.3(7)M
  - Cisco NX-OS Release 10.4(5)M
  - Cisco NX-OS Release 10.5(3)F
- You must enable the VLAN interface feature before the device can apply this functionality.
- Shut down the VLAN network interface for all VLANs that you plan to configure as secondary VLANs before you configure these VLANs.
- When a static MAC is created on a regular VLAN and then that VLAN is converted to a secondary VLAN, the Cisco NX-OS maintains the MAC that was configured on the secondary VLAN as the static MAC.

- PVLANs support PVLAN port modes as follows:
  - Promiscuous.
  - Promiscuous trunk.
  - Isolated host.
  - Isolated host trunk.
  - Community host.
- Beginning with Cisco NX-OS Release 9.2(1), PVLANs support VXLANs.
- Private VLANs provide port mode support for port channels.
- Private VLANs provide port mode support for virtual port channels (vPCs) interfaces.
- When you configure PVLAN promiscuous trunks or PVLAN isolated trunks, we recommend that you allow non-PVLANs in the list specified by the **switchport private-vlan trunk allowed** *id* command. PVLANs are mapped or associated depending on the PVLAN trunk mode.

Note You cannot connect a second switch to a promiscuous or isolated PVLAN trunk. The PVLAN promiscuous trunk or PVLAN isolated trunk is supported only on host-switch.

- The **system private-vlan fex trunk** command is not supported on Cisco Nexus 9300-FX, 9300-FX2, and 9300-FX3 platform switches. In a single-homed FEX configuration, FEX ports support the following PVLAN modes:
  - Isolated host
  - Community host
  - Isolated trunk

  These modes are not supported on FEX port channels or in FEX active-active (AA) or ST vPC configurations.
- PVLANs support PACLs and RACLs.
- PVLANs support SVIs as follows:
  - SVIs on the primary VLANs.
  - Primary and secondary IP addresses on the SVI.
  - HSRP on the primary SVI.
- PVLANs support Layer 2 forwarding.
- PVLANs support STP as follows:
  - RSTPs
  - MSTs

- PVLANs are supported across switches through a regular trunk port.
- PVLANs are supported on the 10G ports of the Cisco Nexus 9396PQ and 93128TX switches.
- PVLAN configurations are not supported on the ALE ports of Cisco Nexus 9300 Series switches.
- PVLAN port mode is not supported on the Cisco Nexus 3164Q switch.
- On Network Forwarding Engines (NFE), PVLANs do not provide support on breakout.
- PVLANs are not supported on vPC or port channel FEX ports.
- PVLANs do not provide support for IP multicast or IGMP snooping.
- Beginning with Cisco NX-OS Release 9.3(3), the following features are supported on Cisco Nexus C9316D-GX, C93600CD-GX, and 9364C-GX switches.
  - vPC
  - 200k Mac scale
  - Dot1x
  - Port-security
  - Selective QinQ
  - Selective QinQ with multiple provider VLAN
- Beginning with Cisco NX-OS Release 9.3(5), PVLANs support DHCP snooping.
- Beginning with Cisco NX-OS Release 9.3(5), PVLAN is supported on N9K-C93180YC-FX3S platform switches.
- Beginning with Cisco NX-OS Release 9.3(9), PVLAN configuration is not allowed on vPC Peer-link interfaces.
- PVLANs do not provide support for PVLAN QoS.
- PVLANs do not provide support for VACLs.
- PVLANs do not provide support for VTP.
- PVLANs do not provide support for tunnels.
- PVLANs do not provide support for SPAN when the source is a PVLAN VLAN.
- You cannot configure a shared interface to be part of a PVLAN. For more details, see the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*.
- Although the Cisco NX-OS CLI allows the configuration of multiple isolated VLAN configurations per PVLAN group, such a configuration is not supported. A PVLAN group can have at most one isolated VLAN.
- PVLAN association on a VLAN is not supported.
- MAC address learning for PVLAN host ports and normal trunks happens on the Primary VLAN. For normal trunks, packets are exchanged using secondary VLAN, but MAC learning is still enforced in Primary VLAN.

- PVLANs are not supported on Cisco Nexus 9500 Series switches with N9K-X9636C-R, N9K-X9636Q-R, N9K-X9636C-RX line cards.
- Beginning with Cisco NX-OS Release 10.1(2), the combination of PVLAN and portSec feature on a vPC orphan port has limitations on dynamic Mac syncing across peers and triggers.
- Beginning with Cisco NX-OS Release 10.2(2)F, the following features are supported on Cisco N9K-9332D-GX2B platform switches.
  - PVLAN and Flex Links
  - VPC
  - Selective QinQ
  - Selective QinQ with multiple provider Vlan
- Beginning with Cisco NX-OS Release 10.2(3)F, if the global command, **system dot1q-tunnel transit**, is configured on the Nexus switch that acts as a transit box, then when a packet comes in with two or more tags, the Private VLAN with Inner VLAN Tag Preservation feature allows for preservation of the inner tag for PVLAN. This feature is supported only on EX, FX, FX2, FX3, GX, and GX2B based Cisco Nexus 9000 Series TOR switches.
- Inner tag preservation does not work when PVLAN and Q-in-Q are configured on the same port.
- Beginning with Cisco NX-OS Release 10.3(3)F, IPv6 underlay is supported on PVLAN for VXLAN EVPN on Cisco Nexus 9300-EX/FX/FX2/FX3/GX/GX2 switches and Cisco Nexus 9500 switches with 9700-EX/FX/GX line cards.
- Beginning with Cisco NX-OS Release 10.4(2)F, you can connect a second switch to a promiscuous trunk or isolated PVLAN trunk ports.
- Beginning with Cisco NX-OS Release 10.4(2)F, PVLAN is supported on C93108TC-FX3 switch.

Default Settings for Private VLANs

This table lists the default setting for private VLANs.

Table 6: Default Private VLAN Setting

Parameters	Default
Private VLANs	Disabled

Configuring a Private VLAN

You must have already created the VLAN before you can assign the specified VLAN as a private VLAN.

See the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*, for information on assigning IP addresses to VLAN interfaces.

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

Enabling Private VLANs - CLI Version

You must enable private VLANs on the device to have the private VLAN functionality.

Note The private VLAN commands do not appear until you enable the private VLAN feature.

SUMMARY STEPS

1. **config t**
2. **feature private-vlan**
3. **exit**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**feature private-vlan**
	Example:
	`switch(config)# feature private-vlan`
	`switch(config)#`
		Enables private VLAN functionality on the device.

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

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

Example

This example shows how to enable private VLAN functionality on the device:

```
switch# config t
switch(config)# feature private-vlan
switch(config)#
```

Configuring a VLAN as a Private VLAN - CLI Version

Note Before you configure a VLAN as a secondary VLAN—that is, either a community or isolated VLAN—you must first shut down the VLAN network interface.

You can configure a VLAN as a private VLAN.

To create a private VLAN, you first create a VLAN and then configure that VLAN to be a private VLAN.

You create all VLANs that you want to use in the private VLAN as a primary VLAN, a community VLAN, or an isolated VLAN. You will later associate multiple isolated and multiple community VLANs to one primary VLAN. You can have many primary VLANs and associations, which means that you could have many private VLANs.

If you delete either the primary or secondary VLAN, the ports that are associated with the VLAN become inactive.

On private VLAN trunk ports, if you delete either the secondary or primary VLAN, only that specific VLAN becomes inactive; the trunk ports stay up.

SUMMARY STEPS

1. **config t**
2. **vlan** {*vlan-id* | *vlan-range*}
3. [**no**] **private-vlan** {**community** | **isolated | primary**}
4. **exit**
5. (Optional) **show vlan private-vlan** [*type*]
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

	Command or Action	Purpose

Step 2	**vlan** {*vlan-id* | *vlan-range*}
	Example:
	`switch(config)# vlan 5`
	`switch(config-vlan)#`
		Places you into the VLAN configuration submode.

Step 3	[**no**] **private-vlan** {**community** | **isolated** | **primary**} **Example:** `switch(config-vlan)# private-vlan primary`	Configures the VLAN as either a community, isolated, or primary private VLAN. In a private VLAN, you must have one primary VLAN. You can have multiple community and isolated VLANs. or Removes the private VLAN configuration from the specified VLAN(s) and returns it to normal VLAN mode. If you delete either the primary or secondary VLAN, the ports that are associated with the VLAN become inactive.

Step 4	**exit**
	Example:
	`switch(config-vlan)# exit`
	`switch(config)#`
		Exits the VLAN configuration submode.

Step 5	(Optional) **show vlan private-vlan** [*type*] **Example:** `switch# show vlan private-vlan`	Displays the private VLAN configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to assign VLAN 5 to a private VLAN as the primary VLAN:

```
switch# config t
switch(config)# vlan 5
switch(config-vlan)# private-vlan primary
switch(config-vlan)# exit
switch(config)#
```

Associating Secondary VLANs with a Primary Private VLAN - CLI Version

Follow these guidelines when you associate secondary VLANs with a primary VLAN:

- The *secondary-vlan-list* parameter cannot contain spaces. It can contain multiple comma-separated items. Each item can be a single secondary VLAN ID or a hyphenated range of secondary VLAN IDs.
- The *secondary-vlan-list* parameter can contain multiple community and isolated VLAN IDs.

- Enter a *secondary-vlan-list* or enter the **add** keyword with a *secondary-vlan-list* to associate secondary VLANs with a primary VLAN.
- Enter the **remove** keyword with a *secondary-vlan-list* to clear the association between secondary VLANs and a primary VLAN.
- You change the association between a secondary and primary VLAN by removing the existing association and then adding the desired association.

If you delete either the primary or secondary VLAN, the ports that are associated with the VLAN become inactive.

When you enter the **no private-vlan** command, the VLAN returns to the normal VLAN mode. All associations on that VLAN are suspended, but the interfaces remain in private VLAN mode.

When you reconvert the specified VLAN to private VLAN mode, the original associations are reinstated.

If you enter the **no vlan** command for the primary VLAN, all private VLAN associations with that VLAN are lost. However, if you enter the **no vlan** command for a secondary VLAN, the private VLAN associations with that VLAN are suspended and return when you recreate the specified VLAN and configure it as the previous secondary VLAN.

Before you begin

Ensure that the private VLAN feature is enabled.

SUMMARY STEPS

1. **config t**
2. **vlan** *primary-vlan-id*
3. [**no**] **private-vlan association** {[**add**] *secondary-vlan-list* | **remove** *secondary-vlan-list*}
4. **exit**
5. (Optional) **show vlan private-vlan** [**type**]
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**vlan** *primary-vlan-id*
	Example:
	`switch(config)# vlan 5`
	`switch(config-vlan)#`
		Enters the number of the primary VLAN that you are working in for the private VLAN configuration.

	Command or Action	Purpose
Step 3	[**no**] **private-vlan association** {[**add**] *secondary-vlan-list* | **remove** *secondary-vlan-list*} **Example:** `switch(config-vlan)# private-vlan association 100-105,109`	Use one form of the command to Associate the secondary VLANs with the primary VLAN. or Remove all associations from the primary VLAN and return it to normal VLAN mode.

Step 4	**exit**
	Example:
	`switch(config-vlan)# exit`
	`switch(config)#`
		Exits the VLAN configuration submode.

Step 5	(Optional) **show vlan private-vlan** [**type**] **Example:** `switch# show vlan private-vlan`	Displays the private VLAN configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to associate community VLANs 100 through 105 and isolated VLAN 109 with primary VLAN 5:

```
switch(config)# vlan 5
switch(config-vlan)# private-vlan association 100-105, 109
switch(config-vlan)# exit
switch(config)#
```

Mapping Secondary VLANs to the VLAN Interface of a Primary VLAN - CLI Version

Note See the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*, for information on assigning IP addresses to VLAN interfaces on primary VLANs of private VLANs.

You map secondary VLANs to the VLAN interface of a primary VLAN. Isolated and community VLANs are both called secondary VLANs. To allow Layer 3 processing of private VLAN ingress traffic, you map secondary VLANs to the VLAN network interface of a primary VLAN.

Note You must enable VLAN network interfaces before you configure the VLAN network interface. VLAN network interfaces on community or isolated VLANs that are associated with a primary VLAN will be out of service. Only the VLAN network interface on the primary VLAN is in service.

Before you begin

- Enable the private VLAN feature.
- Enable the VLAN interface feature.
- Ensure that you are working on the correct primary VLAN Layer 3 interface to map the secondary VLANs.

SUMMARY STEPS

1. **config t**
2. **interface vlan** *primary-vlan-ID*
3. [**no**] **private-vlan mapping** {[**add**] *secondary-vlan-list* | **remove** *secondary-vlan-list*}
4. **exit**
5. (Optional) **show interface vlan** *primary-vlan-id* **private-vlan mapping**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**interface vlan** *primary-vlan-ID*
	Example:
	`switch(config)# interface vlan 5`
	`switch(config-if)#`
		Enters the number of the primary VLAN that you are working in for the private VLAN configuration. Places you into the interface configuration mode for the primary VLAN.

Step 3	[**no**] **private-vlan mapping** {[**add**] *secondary-vlan-list* | **remove** *secondary-vlan-list*} **Example:** `switch(config-if)# private-vlan mapping 100-105, 109`	Map the secondary VLANs to the SVI or Layer 3 interface of the primary VLAN. This action allows the Layer 3 switching of private VLAN ingress traffic. or Clear the mapping to the Layer 3 interface between the secondary VLANs and the primary VLANs.

	Command or Action	Purpose

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

Step 5	(Optional) **show interface vlan** *primary-vlan-id* **private-vlan mapping** **Example:** `switch(config)# show interface vlan 101 private-vlan mapping`	Displays the interface private VLAN information.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to map the secondary VLANs 100 through 105 and 109 on the Layer 3 interface of the primary VLAN 5:

```
switch #config t
switch(config)# interface vlan 5
switch(config-if)# private-vlan mapping 100-105, 109
switch(config-if)# exit
switch(config)#
```

Configuring a Layer 2 Interface as a Private VLAN Host Port

You can configure a Layer 2 interface as a private VLAN host port. In private VLANs, host ports are part of the secondary VLANs, which are either community VLANs or isolated VLANs.

Note We recommend that you enable BPDU Guard on all interfaces configured as a host port.

You then associate the host port with both the primary and secondary VLANs.

Before you begin

Ensure that the private VLAN feature is enabled.

SUMMARY STEPS

1. **config t**
2. **interface** *type slot/port*
3. **switchport mode private-vlan host**
4. [**no**] **switchport private-vlan host-association** {*primary-vlan-id*} {*secondary-vlan-id*}

5. **exit**
6. (Optional) **show interface switchport**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Selects the Layer 2 port to configure as a private VLAN host port.

Step 3	**switchport mode private-vlan host**
	Example:
	`switch(config-if)# switchport mode private-vlan host`
	`switch(config-if)#`
		Configures the Layer 2 port as a host port for a private VLAN.

Step 4	[**no**] **switchport private-vlan host-association** {*primary-vlan-id*} {*secondary-vlan-id*} **Example:** `switch(config-if)# switchport private-vlan host-association 10 50`	Associate the Layer 2 host port with the primary and secondary VLANs of a private VLAN. The secondary VLAN can be either an isolated or community VLAN. or Remove the private VLAN association from the port.

Step 5	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface configuration mode.

Step 6	(Optional) **show interface switchport** **Example:** `switch# show interface switchport`	Displays information on all interfaces configured as switch ports.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the Layer 2 port 2/1 as a host port for a private VLAN and associate it to primary VLAN 10 and secondary VLAN 50:

```
switch# config t
switch(config)# interface ethernet 2/1
switch(config-if)# switchport mode private-vlan host
switch(config-if)# switchport private-vlan host-association 10 50
switch(config-if)# exit
switch(config)#
```

Configuring a Layer 2 Interface as a Private VLAN Isolated Trunk Port

You can configure a Layer 2 interface as a private VLAN isolated trunk port. These isolated trunk ports carry traffic for multiple secondary VLANs as well as normal VLANs.

Note You must associate the primary and secondary VLANs before they become operational on the private VLAN isolated trunk port.

Before you begin

Ensure that the private VLAN feature is enabled.

SUMMARY STEPS

1. **config t**
2. **interface** {*type slot/port*}
3. **switchport**
4. **switchport mode private-vlan trunk secondary**
5. (Optional) **switchport private-vlan trunk native vlan** *vlan-id*
6. **switchport private-vlan trunk allowed vlan** {**add** *vlan-list* | **all** | **except** *vlan-list* | **none** | **remove** *vlan-list*}
7. [**no**] **switchport private-vlan association trunk** {*primary-vlan-id* [*secondary-vlan-id*]}
8. **exit**
9. (Optional) **show interface switchport**
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**config t** **Example:**	Enters configuration mode.

	Command or Action	Purpose
	`switch# config t` `switch(config)#`	

Step 2	**interface** {*type slot/port*}
	Example:
	`switch(config)# interface ethernet 2/11`
	`switch(config-if)#`
		Selects the Layer 2 port to configure as a private VLAN isolated trunk port.

Step 3	**switchport**
	Example:
	`switch(config-if)# switchport`
	`switch(config-if)#`
		Configures the Layer 2 port as a switch port.

Step 4	**switchport mode private-vlan trunk secondary**
	Example:
	`switch(config-if)# switchport mode private-vlan trunk secondary`
	`switch(config-if)#`
		Configures the Layer 2 port as an isolated trunk port to carry traffic for multiple isolated VLANs.
		Note
		You cannot put community VLANs into the isolated trunk port.

Step 5	(Optional) **switchport private-vlan trunk native vlan** *vlan-id* **Example:** `switch(config-if)# switchport private-vlan trunk native vlan 5`	Sets the native VLAN for the 802.1Q trunk. Valid values are from 1 to 3968 and 4048 to 4093. The default value is 1. **Note** If you are using a private VLAN as the native VLAN for the isolated trunk port, you must enter a value for a secondary VLAN or a normal VLAN; you cannot configure a primary VLAN as the native VLAN.

Step 6	**switchport private-vlan trunk allowed vlan** {**add** *vlan-list* | **all** | **except** *vlan-list* | **none** | **remove** *vlan-list*}
	Example:
	`switch(config-if)# switchport private-vlan trunk allowed vlan add 1`
	`switch(config-if)#`
		Sets the allowed VLANs for the private VLAN isolated trunk interface. Valid values are from 1 to 3968 and 4048 to 4093.
		When you map the private primary and secondary VLANs to the isolated trunk port, the system automatically puts all the primary VLANs into the allowed VLAN list for this port.
		Note
		Ensure that the native VLAN is part of the allowed VLAN list. The default for this command is to allow no VLANs on this interface, so you must configure the native VLAN as an allowed VLAN, unless it is already added as an associated VLAN, to pass native VLAN traffic.

Step 7	[**no**] **switchport private-vlan association trunk** {*primary-vlan-id* [*secondary-vlan-id*]} **Example:**	Associate the Layer 2 isolated trunk port with the primary and secondary VLANs of private VLANs. The secondary VLAN must be an isolated VLAN. You can associate a maximum of 16 private VLAN primary and secondary

	Command or Action	Purpose
	`switch(config-if)# switchport private-vlan association trunk 10 101` `switch(config-if)#`	pairs on each isolated trunk port. You must reenter the command for each pair of primary and secondary VLANs that you are working with. **Note** Each secondary VLAN on an isolated trunk port must be associated with a different primary VLAN. You cannot put two isolated VLANs that are associated with the same primary VLAN into a private VLAN isolated trunk port. If you do, the last entry overwrites the previous entry. or Remove the private VLAN association from the private VLAN isolated trunk port.

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

Step 9	(Optional) **show interface switchport** **Example:** `switch# show interface switchport`	Displays information on all interfaces configured as switch ports.
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the Layer 2 port 2/1 as a private VLAN isolated trunk port associated with three different primary VLANs and an associated secondary VLAN:

```
switch# config t
switch(config)# interface ethernet 2/1
switch(config-if)# switchport mode private-vlan trunk
switch(config-if)# switchport private-vlan trunk allowed vlan add 1
switch(config-if)# switchport private-vlan association trunk 10 101
switch(config-if)# switchport private-vlan association trunk 20 201
switch(config-if)# switchport private-vlan association trunk 30 102
switch(config-if)# exit
switch(config)#
```

Configuring a Layer 2 Interface as a Private VLAN Promiscuous Port

You can configure a Layer 2 interface as a private VLAN promiscuous port and then associate that promiscuous port with the primary and secondary VLANs.

Before you begin

Ensure that the private VLAN feature is enabled.

SUMMARY STEPS

1. **config t**
2. **interface** {*type slot/port*}
3. **switchport mode private-vlan promiscuous**
4. [**no**] **switchport private-vlan mapping** {*primary-vlan-id*} {*secondary-vlan-list* | **add** *secondary-vlan-list* | **remove** *secondary-vlan-list*}
5. **exit**
6. (Optional) **show interface switchport**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**interface** {*type slot/port*}
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Selects the Layer 2 port to configure as a private VLAN promiscuous port.

Step 3	**switchport mode private-vlan promiscuous** **Example:** `switch(config-if)# switchport mode private-vlan promiscuous`	Configures the Layer 2 port as a promiscuous port for a private VLAN.
Step 4	[**no**] **switchport private-vlan mapping** {*primary-vlan-id*} {*secondary-vlan-list* | **add** *secondary-vlan-list* | **remove** *secondary-vlan-list*} **Example:** `switch(config-if)# switchport private-vlan mapping 10 50`	Configure the Layer 2 port as a promiscuous port and associates the specified port with a primary VLAN and a selected list of secondary VLANs. The secondary VLAN can be either an isolated or community VLAN. or Clear the mapping from the private VLAN.

Step 5	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits the interface configuration mode.

	Command or Action	Purpose
Step 6	(Optional) **show interface switchport** **Example:** `switch# show interface switchport`	Displays information on all interfaces configured as switch ports.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the Layer 2 port 2/1 as a promiscuous port associated with the primary VLAN 10 and the secondary isolated VLAN 50:

```
switch# config t
switch(config)# interface ethernet 2/1
switch(config-if)# switchport mode private-vlan promiscuous
switch(config-if)# switchport private-vlan mapping 10 50
switch(config-if)# exit
switch(config)#
```

Configuring a Layer 2 Interface as a Private VLAN Promiscuous Trunk Port

You can configure a Layer 2 interface as a private VLAN promiscuous trunk port and then associate that promiscuous trunk port with multiple primary VLANs. These promiscuous trunk ports carry traffic for multiple primary VLANs as well as normal VLANs.

Note You must associate the primary and secondary VLANs before they become operational on the private VLAN promiscuous trunk port.

Before you begin

Ensure that the private VLAN feature is enabled.

SUMMARY STEPS

1. **config t**
2. **interface** {*type slot/port*}
3. **switchport**
4. **switchport mode private-vlan trunk promiscuous**
5. (Optional) **switchport private-vlan trunk native vlan** *vlan-id*
6. **switchport mode private-vlan trunk allowed vlan** {**add** *vlan-list* | **all** | **except** *vlan-list* | **none** | **remove** *vlan-list*}
7. [**no**]**switchport private-vlan mapping trunk** *primary-vlan-id* [*secondary-vlan-id*] {**add** *secondary-vlan-list* | **remove** *secondary-vlan-id*}

8. **exit**
9. (Optional) **show interface switchport**
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**interface** {*type slot/port*}
	Example:
	`switch(config)# interface ethernet 2/1`
	`switch(config-if)#`
		Selects the Layer 2 port to configure as a private VLAN promiscuous trunk port.

Step 3	**switchport**
	Example:
	`switch(config-if)# switchport`
	`switch(config-if)#`
		Configures the Layer 2 port as a switch port.

Step 4	**switchport mode private-vlan trunk promiscuous**
	Example:
	`switch(config-if)# switchport mode private-vlan trunk promiscuous`
	`switch(config-if)#`
		Configures the Layer 2 port as a promiscuous trunk port to carry traffic for multiple private VLANs as well as normal VLANs.

Step 5	(Optional) **switchport private-vlan trunk native vlan** *vlan-id* **Example:** `switch(config-if)# switchport private-vlan trunk native vlan 5`	Sets the native VLAN for the 802.1Q trunk. Valid values are from 1 to 3968 and 4048 to 4093. The default value is 1. **Note** If you are using a private VLAN as the native VLAN for the promiscuous trunk port, you must enter a value for a primary VLAN or a normal VLAN; you cannot configure a secondary VLAN as the native VLAN.

Step 6	**switchport mode private-vlan trunk allowed vlan** {**add** *vlan-list* | **all** | **except** *vlan-list* | **none** | **remove** *vlan-list*}
	Example:
	`switch(config-if)# switchport private-vlan trunk allowed vlan add 1`
	`switch(config-if)#`
		Sets the allowed VLANs for the private VLAN promiscuous trunk interface. Valid values are from 1 to 3968 and 4048 to 4093.
		When you map the private primary and secondary VLANs to the promiscuous trunk port, the system automatically puts all the primary VLANs into the allowed VLAN list for this port.
		Note

	Command or Action	Purpose
		Ensure that the native VLAN is part of the allowed VLAN list. The default for this command is to allow no VLANs on this interface, so you must configure the native VLAN as an allowed VLAN, unless it is already added as an associated VLAN, to pass native VLAN traffic.

Step 7	[**no**]**switchport private-vlan mapping trunk** *primary-vlan-id* [*secondary-vlan-id*] {**add** *secondary-vlan-list* | **remove** *secondary-vlan-id*}
	Example:
	`switch(config-if)# switchport private-vlan mapping trunk 4 5`
	`switch(config-if)#`
		Map or remove the mapping for the promiscuous trunk port with the primary VLAN and a selected list of associated secondary VLANs. The secondary VLAN can be either an isolated or community VLAN. The private VLAN association between primary and secondary VLANs must be operational to pass traffic. You can map a maximum of 16 private VLAN primary and secondary pairs on each promiscuous trunk port. You must reenter the command for each primary VLAN that you are working with.
		or
		Remove the private VLAN promiscuous trunk mappings from the interface.

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

Step 9	(Optional) **show interface switchport** **Example:** `switch# show interface switchport`	Displays information on all interfaces configured as switch ports.
Step 10	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the Layer 2 port 2/1 as a promiscuous trunk port associated with two primary VLANs and their associated secondary VLANs:

```
switch# config t
switch(config)# interface ethernet 2/1
switch(config-if)# switchport
switch(config-if)# switchport mode private-vlan trunk promiscuous
switch(config-if)# switchport private-vlan trunk allowed vlan add 1
switch(config-if)# switchport private-vlan mapping trunk 10 20
switch(config-if)# switchport private-vlan mapping trunk 11 21
switch(config-if)# exit
switch(config)#
```

Verifying the Private VLAN Configuration

To display private VLAN configuration information, perform one of the following tasks:

Command	Purpose
show running-config vlan *vlan-id*	Displays VLAN information.
show vlan private-vlan [*type*]	Displays information on private VLANs.
show interface private-vlan mapping	Displays information on interfaces for private VLAN mapping.
show interface vlan *primary-vlan-id* **private-vlan mapping**	Displays information on interfaces for private VLAN mapping.
show interface switchport	Displays information on all interfaces configured as switch ports.

Displaying and Clearing Private VLAN Statistics

To display private VLAN configuration information, perform one of the following tasks:

Command	Purpose
clear vlan [**id** *vlan-id*] **counters**	Clears counters for all VLANs or for a specified VLAN.
show vlan counters	Displays information on Layer 2 packets in each VLAN.

Configuration Examples for Private VLANs

The following example shows how to create the three types of private VLANs, how to associate the secondary VLANs to the primary VLAN, how to create a private VLAN host and promiscuous port and assign them to the correct VLAN, and how to create a VLAN interface, or SVI, to allow the primary VLAN to communicate with the rest of the network:

```
switch# configure terminal
switch(config)# vlan 2
switch(config-vlan)# private-vlan primary
switch(config-vlan)# exit
switch(config)# vlan 3
switch(config-vlan)# private-vlan community
switch(config-vlan)# exit
switch(config)# vlan 4
switch(config-vlan)# private-vlan isolated
switch(config-vlan)# exit

switch(config)# vlan 2
```

```
switch(config-vlan)# private-vlan association 3,4
switch(config-vlan)# exit

switch(config)# interface ethernet 1/11
switch(config-if)# switchport
switch(config-if)# switchport mode private-vlan host
switch(config-if)# exit
switch(config)# interface ethernet 1/12
switch(config-if)# switchport
switch(config-if)# switchport mode private-vlan promiscuous
switch(config-if)# exit

switch(config)# interface ethernet 1/11
switch(config-if)# switchport private-vlan host-association 2 3
switch(config-if)# exit
switch(config)# interface ethernet 1/12
switch(config-if)# switchport private-vlan mapping 2 3,4
switch(config-if)# exit

switch(config)# interface vlan 2
switch(config-vlan)# private-vlan mapping 3,4
switch(config-vlan)# exit
switch(config)#
```

Additional References for Private VLANs -- CLI Version

Related Documents

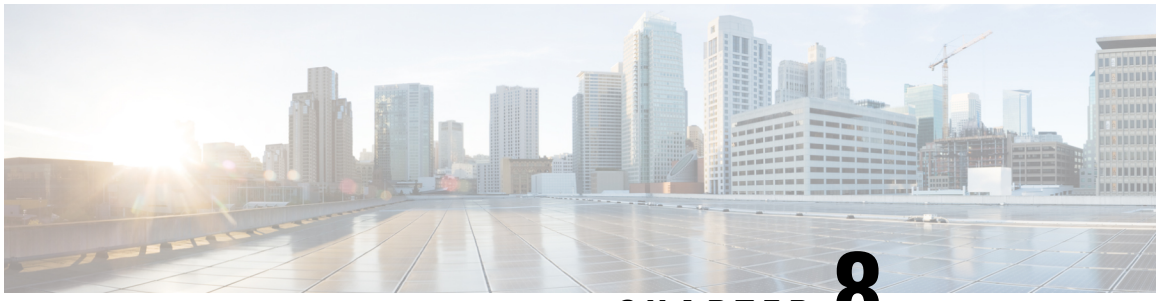
Related Topic	Document Title
VLAN interfaces, IP addressing	*Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*
Static MAC addresses, security	*Cisco Nexus 9000 Series NX-OS Security Configuration Guide*
Cisco NX-OS fundamentals	*Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide*
High availability	*Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide*
System management	*Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*
Licensing	*Cisco NX-OS Licensing Guide*
Release notes	*Cisco Nexus 9000 Series NX-OS Release Notes*

Standards

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

MIBs

MIBs	MIBs Link
• CISCO-PRIVATE-VLAN-MIB	For more information, refer to https://cisco.github.io/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html.



CHAPTER 8

Configuring Switching Modes

- Information About Switching Modes, on page 93
- Guidelines and Limitations for Switching Modes, on page 93
- Default Settings for Switching Modes, on page 94
- Configuring Switching Modes, on page 94

Information About Switching Modes

The switching mode determines whether the switch begins forwarding the frame as soon as the switch has read the destination details in the packet header or waits until the entire frame has been received and checked for cyclic redundancy check (CRC) errors before forwarding them to the network.

The switching mode is applicable to all packets being switched or routed through the hardware and can be saved persistently through reboots and restarts.

The switch operates in either of the following switching modes:

Cut-Through Switching Mode

Cut-through switching mode is enabled by default. Switches operating in cut-through switching mode start forwarding the frame as soon as the switch has read the destination details in the packet header. A switch in cut-through mode forwards the data before it has completed receiving the entire frame.

The switching speed in cut-through mode is faster than the switching speed in store-and-forward switching mode.

Store-and-Forward Switching Mode

When store-and-forward switching is enabled, the switch checks each frame for cyclic redundancy check (CRC) errors before forwarding them to the network. Each frame is stored until the entire frame has been received and checked.

Because it waits to forward the frame until the entire frame has been received and checked, the switching speed in store-and-forward switching mode is slower than the switching speed in cut-through switching mode.

Guidelines and Limitations for Switching Modes

Consider the following guidelines and limitations for each of the switching modes:

Cut-Through Switching Mode Guidelines and Limitations

- **show** commands with the **internal** keyword are not supported.
- Packets with FCS errors are not mirrored if SPAN is configured.

Store-and-Forward Switching Mode Guidelines and Limitations

- **show** commands with the **internal** keyword are not supported.
- Packets with FCS errors are dropped.
- Packets with FCS errors are not mirrored if SPAN is configured.
- The CPU port always operates in store-and-forward mode. Any packets forwarded to the CPU with FCS errors are dropped.
- Store-and-forward mode activates automatically for a port when the switch identifies that the port is oversubscribed and the ingress rate is greater than the switching capacity of the egress port. For example, when the port ingress rate is 10 gigabit and the switching capacity of the egress port is 1 gigabit.

Note The global configuration does not change, even if store-and-forward mode is activated for an oversubscribed port.

Switching mode value added to SNMP MIB

Beginning with Cisco NX-OS Release 10.5(3)F, switching mode is now available with SNMP **CISCO-SYSTEM-EXT-MIB**. You can use the SNMP **get** command to view the switching mode.

In releases before Cisco NX-OS Release 10.5(3)F, the switching mode property in the DME database is set to **Default** when the **no switching-mode store-forward** command is configured on the switch.

Starting with Cisco NX-OS Release 10.5(3)F, the switching mode property in the DME database is set to **Cut-through** when the **no switching-mode store-forward** command is configured on the switch.

Default Settings for Switching Modes

Cut-through switching is enabled by default.

Configuring Switching Modes

Enabling Store-and-Forward Switching

Note Enabling store-and-forward switching mode might impact your port-to-port switching latency.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config) # **switching-mode store-forward**
3. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config) # **switching-mode store-forward**	Enables store-and-forward switching mode.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable store-and-forward switching:

```
switch# configure terminal
switch(config) # switching-mode store-forward
switch(config) #
```

Reenabling Cut-Through Switching

Cut-through switching is enabled by default. To reenable cut-through switching, use the **no** form of the **switching-mode store-forward** command.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config) # **no switching-mode store-forward**
3. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config) # **no switching-mode store-forward**	Disables store-and-forward switching mode. Enables cut-through switching mode.

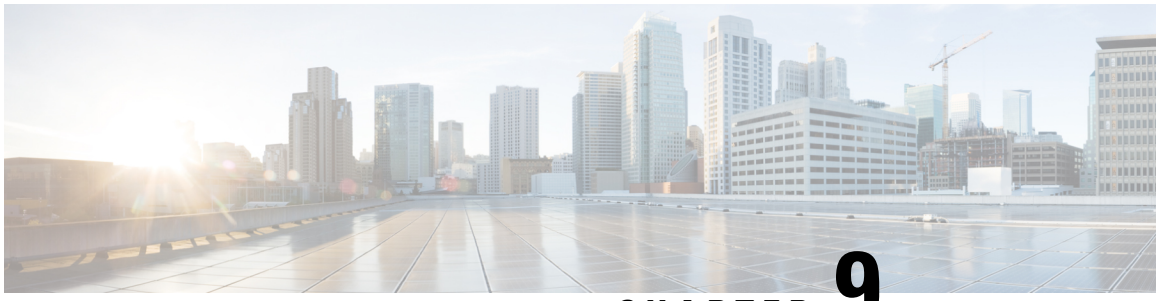
	Command or Action	Purpose
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to reenable cut-through switching:

```
switch# configure terminal
switch(config) # no switching-mode store-forward
switch(config) #
```

Note The command **no switching-mode store-forward** is not supported on Cisco Nexus 9800 Series switches as Cut-Through mode is not available on this platform.



CHAPTER 9

Configuring Rapid PVST+ Using Cisco NX-OS

- Information About Rapid PVST+, on page 97
- Prerequisites for Configuring Rapid PVST+, on page 113
- Guidelines and Limitations for Configuring Rapid PVST+, on page 113
- Default Settings for Rapid PVST+, on page 114
- Configuring Rapid PVST+, on page 115
- Verifying the Rapid PVST+ Configurations, on page 131
- Displaying and Clearing Rapid PVST+ Statistics -- CLI Version, on page 132
- Rapid PVST+ Example Configurations, on page 132
- Additional References for Rapid PVST+ -- CLI Version, on page 132

Information About Rapid PVST+

Note See the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*, for information on creating Layer 2 interfaces.

The Spanning Tree Protocol (STP) was implemented to provide a loop-free network at Layer 2 of the network. Rapid PVST+ is an updated implementation of STP that allows you to create one spanning tree topology for each VLAN. Rapid PVST+ is the default STP mode on the device.

Note Spanning tree is used to refer to IEEE 802.1w and IEEE 802.1s. If the IEEE 802.1D Spanning Tree Protocol is discussed in this publication, then 802.1D is stated specifically.

Note Rapid PVST+ is the default STP mode.

The Rapid PVST+ protocol is the IEEE 802.1w standard, Rapid Spanning Tree Protocol (RSTP), implemented on a per VLAN basis. Rapid PVST+ interoperates with the IEEE 802.1Q VLAN standard, which mandates a single STP instance for all VLANs, rather than per VLAN.

Rapid PVST+ is enabled by default on the default VLAN (VLAN1) and on all newly created VLANs on the device. Rapid PVST+ interoperates with devices that run legacy IEEE 802.1D STP.

RSTP is an improvement on the original STP standard, 802.1D, which allows faster convergence.

Note The device supports full nondisruptive upgrades for Rapid PVST+. See the Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide, for complete information on nondisruptive upgrades.

STP

STP is a Layer 2 link-management protocol that provides path redundancy while preventing loops in the network.

Overview of STP

In order for a Layer 2 Ethernet network to function properly, only one active path can exist between any two stations. STP operation is transparent to end stations, which cannot detect whether they are connected to a single LAN segment or a switched LAN of multiple segments.

When you create fault-tolerant internetworks, you must have a loop-free path between all nodes in a network. The STP algorithm calculates the best loop-free path throughout a switched Layer 2 network. Layer 2 LAN ports send and receive STP frames, which are called Bridge Protocol Data Units (BPDUs), at regular intervals. Network devices do not forward these frames but use the frames to construct a loop-free path.

Multiple active paths between end stations cause loops in the network. If a loop exists in the network, end stations might receive duplicate messages and network devices might learn end station MAC addresses on multiple Layer 2 LAN ports.

STP defines a tree with a root bridge and a loop-free path from the root to all network devices in the Layer 2 network. STP forces redundant data paths into a blocked state. If a network segment in the spanning tree fails and a redundant path exists, the STP algorithm recalculates the spanning tree topology and activates the blocked path.

When two Layer 2 LAN ports on a network device are part of a loop, the STP port priority and port path-cost setting determine which port on the device is put in the forwarding state and which port is put in the blocking state. The STP port priority value is the efficiency with which that location allows the port to pass traffic. The STP port path-cost value is derived from the media speed.

How a Topology is Created

All devices in a LAN that participate in a spanning tree gather information about other switches in the network by exchanging BPDUs. This exchange of BPDUs results in the following actions:

- The system elects a unique root switch for the spanning tree network topology.
- The system elects a designated switch for each LAN segment.
- The system eliminates any loops in the switched network by placing redundant switch ports in a backup state; all paths that are not needed to reach the root device from anywhere in the switched network are placed in an STP-blocked state.

The topology on an active switched network is determined by the following:

- The unique device identifier Media Access Control (MAC) address of the device that is associated with each device
- The path cost to the root that is associated with each switch port
- The port identifier that is associated with each switch port

In a switched network, the root switch is the logical center of the spanning tree topology. STP uses BPDUs to elect the root switch and root port for the switched network.

Note The **mac-address bpdu source version 2** command enables STP to use the new Cisco MAC address (00:26:0b:xx:xx:xx) as the source address of BPDUs generated on vPC ports.

To apply this command, you must have identical configurations for both vPC peer switches or peers.

Cisco strongly recommends that you disable ether channel guard on the edge devices before issuing this command to minimize traffic disruption from STP inconsistencies. Re-enable the ether channel guard after updating on both peers.

Bridge ID

Each VLAN on each network device has a unique 64-bit bridge ID that consists of a bridge priority value, an extended system ID (IEEE 802.1t), and an STP MAC address allocation.

Bridge Priority Value

The bridge priority is a 4-bit value when the extended system ID is enabled.

You can only specify a device bridge ID (used by the spanning tree algorithm to determine the identity of the root bridge; the lowest number is preferred) as a multiple of 4096.

Note In this device, the extended system ID is always enabled; you cannot disable the extended system ID.

Extended System ID

The device always uses the 12-bit extended system ID.

Figure 4: Bridge ID with Extended System ID

This figure shows the 12-bit extended system ID field that is part of the bridge ID.

This table shows how the system ID extension combined with the bridge ID functions as the unique identifier for a VLAN.

Table 7: Bridge Priority Value and Extended System ID with the Extended System ID Enabled

Bridge Priority Value				Extended System ID (Set Equal to the VLAN ID)											
Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

STP MAC Address Allocation

Note MAC address reduction is always enabled on the device.

Because MAC address reduction is always enabled on the device, you should also enable MAC address reduction on all other Layer 2 connected network devices to avoid undesirable root bridge election and spanning tree topology issues.

When MAC address reduction is enabled, the root bridge priority becomes a multiple of 4096 plus the VLAN ID. You can only specify a device bridge ID (used by the spanning tree algorithm to determine the identity of the root bridge; the lowest number is preferred) as a multiple of 4096. Only the following values are possible:

- 0
- 4096
- 8192
- 12288
- 16384
- 20480
- 24576
- 28672
- 32768
- 36864
- 40960
- 45056
- 49152
- 53248
- 57344
- 61440

STP uses the extended system ID plus a MAC address to make the bridge ID unique for each VLAN.

Note If another bridge in the same spanning tree domain does not run the MAC address reduction feature, it could win the root bridge ownership because of the finer granularity in the selection of its bridge ID.

BPDUs

Network devices transmit BPDUs throughout the STP instance. Each network device sends configuration BPDUs to communicate and compute the spanning tree topology. Each configuration BPDU contains the following minimal information:

- The unique bridge ID of the network device that the transmitting network device believes to be the root bridge
- The STP path cost to the root
- The bridge ID of the transmitting bridge
- The message age
- The identifier of the transmitting port
- Values for the hello, forward delay, and max-age protocol timer
- Additional information for STP extension protocols

When a network device transmits a Rapid PVST+ BPDU frame, all network devices connected to the VLAN on which the frame is transmitted receive the BPDU. When a network device receives a BPDU, it does not forward the frame but instead uses the information in the frame to calculate a BPDU. If the topology changes, the device initiates a BPDU exchange.

A BPDU exchange results in the following:

- One network device is elected as the root bridge.
- The shortest distance to the root bridge is calculated for each network device based on the path cost.
- A designated bridge for each LAN segment is selected. This network device is closest to the root bridge through which frames are forwarded to the root.
- A root port is elected. This port provides the best path from the bridge to the root bridge.
- Ports included in the spanning tree are selected.

Election of the Root Bridge

For each VLAN, the network device with the lowest numerical ID is elected as the root bridge. If all network devices are configured with the default priority (32768), the network device with the lowest MAC address in the VLAN becomes the root bridge. The bridge priority value occupies the most significant bits of the bridge ID.

When you change the bridge priority value, you change the probability that the device will be elected as the root bridge. Configuring a lower value increases the probability; a higher value decreases the probability.

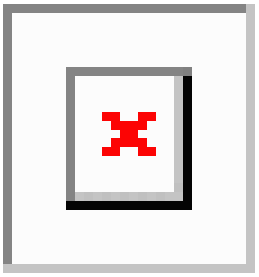
The STP root bridge is the logical center of each spanning tree topology in a Layer 2 network. All paths that are not needed to reach the root bridge from anywhere in the Layer 2 network are placed in STP blocking mode.

BPDUs contain information about the transmitting bridge and its ports, including bridge and MAC addresses, bridge priority, port priority, and path cost. STP uses this information to elect the root bridge for the STP instance, to elect the root port that leads to the root bridge, and to determine the designated port for each Layer 2 segment.

Creating the Spanning Tree Topology

By lowering the numerical value of the ideal network device so that it becomes the root bridge, you force an STP recalculation to form a new spanning tree topology with the ideal network device as the root.

Figure 5: Spanning Tree Topology



In this figure, switch A is elected as the root bridge because the bridge priority of all the network devices is set to the default (32768) and switch A has the lowest MAC address. However, due to traffic patterns, the number of forwarding ports, or link types, switch A might not be the ideal root bridge.

When the spanning tree topology is calculated based on default parameters, the path between the source and destination end stations in a switched network might not be ideal. For instance, connecting higher-speed links to a port that has a higher number than the current root port can cause a root-port change. The goal is to make the fastest link the root port.

For example, assume that one port on switch B is a fiber-optic link, and another port on switch B (an unshielded twisted-pair [UTP] link) is the root port. Network traffic might be more efficient over the high-speed fiber-optic link. By changing the STP port priority on the fiber-optic port to a higher priority (lower numerical value) than the root port, the fiber-optic port becomes the new root port.

Rapid PVST+

Rapid PVST+ is the default spanning tree mode for the software and is enabled by default on the default VLAN and all newly created VLANs.

A single instance, or topology, of RSTP runs on each configured VLAN, and each Rapid PVST+ instance on a VLAN has a single root device. You can enable and disable STP on a per-VLAN basis when you are running Rapid PVST+.

Overview of Rapid PVST+

Rapid PVST+ is the IEEE 802.1w (RSTP) standard implemented per VLAN. A single instance of STP runs on each configured VLAN (if you do not manually disable STP). Each Rapid PVST+ instance on a VLAN has a single root switch. You can enable and disable STP on a per-VLAN basis when you are running Rapid PVST+.

Note Rapid PVST+ is the default STP mode for the device.

Rapid PVST+ uses point-to-point wiring to provide rapid convergence of the spanning tree. The spanning tree reconfiguration can occur in less than 1 second with Rapid PVST+ (in contrast to 50 seconds with the default settings in the 802.1D STP). The device automatically checks the PVID.

Note Rapid PVST+ supports one STP instance for each VLAN.

Using Rapid PVST+, STP convergence occurs rapidly. By default, each designated port in the STP sends out a BPDU every 2 seconds. On a designated port in the topology, if hello messages are missed three consecutive times, or if the maximum age expires, the port immediately flushes all protocol information in the table. A port considers that it loses connectivity to its direct neighbor designated port if it misses three BPDUs or if the maximum age expires. This rapid aging of the protocol information allows quick failure detection.

Rapid PVST+ provides for rapid recovery of connectivity following the failure of a device, a device port, or a LAN. It provides rapid convergence for edge ports, new root ports, and ports connected through point-to-point links as follows:

- Edge ports—When you configure a port as an edge port on an RSTP device, the edge port immediately transitions to the forwarding state. (This immediate transition was previously a Cisco-proprietary feature named PortFast.) You should only configure ports that connect to a single end station as edge ports. Edge ports do not generate topology changes when the link changes.

Enter the **spanning-tree port type** interface configuration command to configure a port as an STP edge port.

Note We recommend that you configure all ports connected to a Layer 2 host as edge ports.

- Root port—If Rapid PVST+ selects a new root port, it blocks the old root port and immediately transitions the new root port to the forwarding state.
- Point-to-point links—If you connect a port to another port through a point-to-point link and the local port becomes a designated port, it negotiates a rapid transition with the other port by using the proposal-agreement handshake to ensure a loop-free topology.

Rapid PVST+ achieves rapid transition to the forwarding state only on edge ports and point-to-point links. Although the link type is configurable, the system automatically derives the link type information from the duplex setting of the port. Full-duplex ports are assumed to be point-to-point ports, while half-duplex ports are assumed to be shared ports.

Edge ports do not generate topology changes, but all other designated and root ports generate a topology change (TC) BPDU when they either fail to receive three consecutive BPDUs from the directly connected neighbor or the maximum age times out. At this point, the designated or root port sends a BPDU with the TC flag set. The BPDUs continue to set the TC flag as long as the TC While timer runs on that port. The value of the TC While timer is the value set for the hello time plus 1 second. The initial detector of the topology change immediately floods this information throughout the entire topology.

When Rapid PVST+ detects a topology change, the protocol does the following:

- Starts the TC While timer with a value equal to twice the hello time for all the nonedge root and designated ports, if necessary.
- Flushes the MAC addresses associated with all these ports.

The topology change notification floods quickly across the entire topology. The system flushes dynamic entries immediately on a per-port basis when it receives a topology change.

Note The TCA flag is used only when the device is interacting with devices that are running legacy 802.1D STP.

The proposal and agreement sequence then quickly propagates toward the edge of the network and quickly restores connectivity after a topology change.

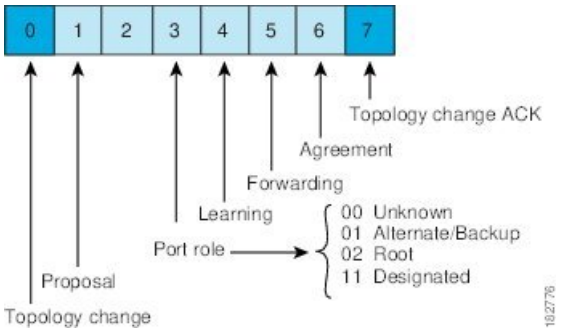
Rapid PVST+ BPDUs

Rapid PVST+ and 802.1w use all six bits of the flag byte to add the following:

- The role and state of the port that originates the BPDU
- The proposal and agreement handshake

Figure 6: Rapid PVST+ Flag Byte in BPDU

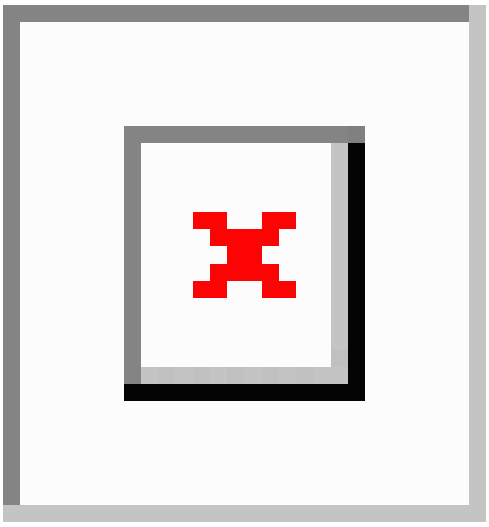
This figure shows the use of the BPDU flags in Rapid PVST+.



Another important change is that the Rapid PVST+ BPDU is type 2, version 2, which makes it possible for the device to detect connected legacy (802.1D) bridges. The BPDU for 802.1D is type 0, version 0.

Proposal and Agreement Handshake

Figure 7: Proposal and Agreement Handshaking for Rapid Convergence



In this figure, switch A is connected to switch B through a point-to-point link, and all of the ports are in the blocking state. Assume that the priority of switch A is a smaller numerical value than the priority of switch B. Switch A sends a proposal message (a configuration BPDU with the proposal flag set) to switch B, proposing itself as the designated switch.

After receiving the proposal message, switch B selects as its new root port the port from which the proposal message was received, forces all nonedge ports to the blocking state, and sends an agreement message (a BPDU with the agreement flag set) through its new root port.

After receiving the agreement message from switch B, switch A also immediately transitions its designated port to the forwarding state. No loops in the network can form because switch B blocked all of its nonedge ports and because there is a point-to-point link between switches A and B.

When switch C connects to switch B, a similar set of handshaking messages are exchanged. Switch C selects the port connected to switch B as its root port, and both ends of the link immediately transition to the forwarding state. With each iteration of this handshaking process, one more switch joins the active topology. As the network converges, this proposal-agreement handshaking progresses from the root toward the leaves of the spanning tree as shown in this figure.

The switch learns the link type from the port duplex mode; a full-duplex port is considered to have a point-to-point connection and a half-duplex port is considered to have a shared connection. You can override the default setting that is controlled by the duplex setting by entering the **spanning-tree link-type** interface configuration command.

This proposal/agreement handshake is initiated only when a nonedge port moves from the blocking to the forwarding state. The handshaking process then proliferates step-by-step throughout the topology.

Protocol Timers

This table describes the protocol timers that affect the Rapid PVST+ performance.

Table 8: Rapid PVST+ Protocol Timers

Variable	Description
Hello timer	Determines how often each device broadcasts BPDUs to other network devices. The default is 2 seconds, and the range is from 1 to 10.
Forward delay timer	Determines how long each of the listening and learning states last before the port begins forwarding. This timer is generally not used by the protocol, but it is used when interoperating with the 802.1D spanning tree. The default is 15 seconds, and the range is from 4 to 30 seconds.
Maximum age timer	Determines the amount of time that protocol information received on a port is stored by the network device. This timer is generally not used by the protocol, but it is used when interoperating with the 802.1D spanning tree. The default is 20 seconds; the range is from 6 to 40 seconds.

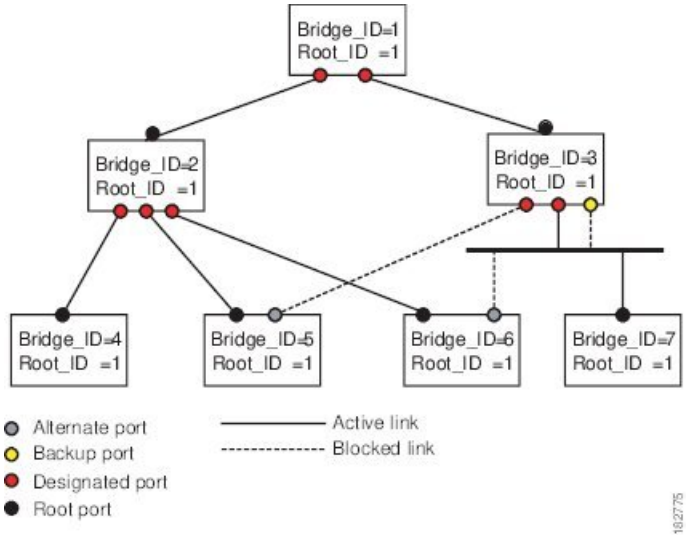
Port Roles

Rapid PVST+ provides rapid convergence of the spanning tree by assigning port roles and learning the active topology. Rapid PVST+ builds upon the 802.1D STP to select the device with the highest switch priority (lowest numerical priority value) as the root bridge. Rapid PVST+ assigns one of these port roles to individual ports:

- Root port—Provides the best path (lowest cost) when the device forwards packets to the root bridge.
- Designated port—Connects to the designated device that has the lowest path cost when forwarding packets from that LAN to the root bridge. The port through which the designated device is attached to the LAN is called the designated port.
- Alternate port—Offers an alternate path toward the root bridge to the path provided by the current root port. An alternate port provides a path to another device in the topology.
- Backup port—Acts as a backup for the path provided by a designated port toward the leaves of the spanning tree. A backup port can exist only when two ports are connected in a loopback by a point-to-point link or when a device has two or more connections to a shared LAN segment. A backup port provides another path in the topology to the device.
- Disabled port—Has no role within the operation of the spanning tree.

In a stable topology with consistent port roles throughout the network, Rapid PVST+ ensures that every root port and designated port immediately transition to the forwarding state while all alternate and backup ports are always in the blocking state. Designated ports start in the blocking state. The port state controls the operation of the forwarding and learning processes.

Figure 8: Sample Topology Demonstrating Port Roles



This figure shows port roles. A port with the root or a designated port role is included in the active topology. A port with the alternate or backup port role is excluded from the active topology.

Rapid PVST+ Port State Overview

Propagation delays can occur when protocol information passes through a switched LAN. As a result, topology changes can take place at different times and at different places in a switched network. When a Layer 2 LAN port transitions directly from nonparticipation in the spanning tree topology to the forwarding state, it can create temporary data loops. Ports must wait for new topology information to propagate through the switched LAN before starting to forward frames.

Each Layer 2 LAN port on the device that uses Rapid PVST+ or MST exists in one of the following four states:

- Blocking—The Layer 2 LAN port does not participate in frame forwarding.
- Learning—The Layer 2 LAN port prepares to participate in frame forwarding.
- Forwarding—The Layer 2 LAN port forwards frames.
- Disabled—The Layer 2 LAN port does not participate in STP and is not forwarding frames.

When you enable Rapid PVST+, every port in the device, VLAN, and network goes through the blocking state and the transitory states of learning at power up. If properly configured, each Layer 2 LAN port stabilizes to the forwarding or blocking state.

When the STP algorithm places a Layer 2 LAN port in the forwarding state, the following process occurs:

1. The Layer 2 LAN port is put into the blocking state while it waits for protocol information that suggests it should go to the learning state.
2. The Layer 2 LAN port waits for the forward delay timer to expire, moves the Layer 2 LAN port to the learning state, and restarts the forward delay timer.
3. In the learning state, the Layer 2 LAN port continues to block frame forwarding as it learns the end station location information for the forwarding database.

4. The Layer 2 LAN port waits for the forward delay timer to expire and then moves the Layer 2 LAN port to the forwarding state, where both learning and frame forwarding are enabled.

Blocking State

A Layer 2 LAN port in the blocking state does not participate in frame forwarding.

A Layer 2 LAN port in the blocking state performs as follows:

- Discards frames received from the attached segment.
- Discards frames switched from another port for forwarding.
- Does not incorporate the end station location into its address database. (There is no learning on a blocking Layer 2 LAN port, so there is no address database update.)
- Receives BPDUs and directs them to the system module.
- Receives, processes, and transmits BPDUs received from the system module.
- Receives and responds to control plane messages.

Learning State

A Layer 2 LAN port in the learning state prepares to participate in frame forwarding by learning the MAC addresses for the frames. The Layer 2 LAN port enters the learning state from the blocking state.

A Layer 2 LAN port in the learning state performs as follows:

- Discards frames received from the attached segment.
- Discards frames switched from another port for forwarding.
- Incorporates the end station location into its address database.
- Receives BPDUs and directs them to the system module.
- Receives, processes, and transmits BPDUs received from the system module.
- Receives and responds to control plane messages.

Forwarding State

A Layer 2 LAN port in the forwarding state forwards frames. The Layer 2 LAN port enters the forwarding state from the learning state.

A Layer 2 LAN port in the forwarding state performs as follows:

- Forwards frames received from the attached segment.
- Forwards frames switched from another port for forwarding.
- Incorporates the end station location information into its address database.
- Receives BPDUs and directs them to the system module.
- Processes BPDUs received from the system module.
- Receives and responds to control plane messages.

Disabled State

A Layer 2 LAN port in the disabled state does not participate in frame forwarding or STP. A Layer 2 LAN port in the disabled state is virtually nonoperational.

A disabled Layer 2 LAN port performs as follows:

- Discards frames received from the attached segment.
- Discards frames switched from another port for forwarding.
- Does not incorporate the end station location into its address database. (There is no learning, so there is no address database update.)
- Does not receive BPDUs from neighbors.
- Does not receive BPDUs for transmission from the system module.

Summary of Port States

This table lists the possible operational and Rapid PVST+ states for ports and whether the port is included in the active topology.

Table 9: Port State Active Topology

Operational Status	Port State	Is Port Included in the Active Topology?
Enabled	Blocking	No
Enabled	Learning	Yes
Enabled	Forwarding	Yes
Disabled	Disabled	No

Synchronization of Port Roles

When the device receives a proposal message on one of its ports and that port is selected as the new root port, Rapid PVST+ forces all other ports to synchronize with the new root information.

The device is synchronized with superior root information received on the root port if all other ports are synchronized. An individual port on the device is synchronized if either of the following applies:

- That port is in the blocking state.
- It is an edge port (a port configured to be at the edge of the network).

If a designated port is in the forwarding state and is not configured as an edge port, it transitions to the blocking state when the Rapid PVST+ forces it to synchronize with new root information. In general, when the Rapid PVST+ forces a port to synchronize with root information and the port does not satisfy any of the above conditions, its port state is set to blocking.

After ensuring that all of the ports are synchronized, the device sends an agreement message to the designated device that corresponds to its root port. When the devices connected by a point-to-point link are in agreement about their port roles, Rapid PVST+ immediately transitions the port states to the forwarding state.

Figure 9: Sequence of Events During Rapid Convergence

This figure shows the sequence of events during synchronization.

Processing Superior BPDU Information

A superior BPDU is a BPDU with root information (such as a lower switch ID or lower path cost) that is superior to what is currently stored for the port.

If a port receives a superior BPDU, Rapid PVST+ triggers a reconfiguration. If the port is proposed and is selected as the new root port, Rapid PVST+ forces all the designated, nonedge ports to synchronize.

If the received BPDU is a Rapid PVST+ BPDU with the proposal flag set, the device sends an agreement message after all of the other ports are synchronized. The new root port transitions to the forwarding state as soon as the previous port reaches the blocking state.

If the superior information received on the port causes the port to become a backup port or an alternate port, Rapid PVST+ sets the port to the blocking state and sends an agreement message. The designated port continues sending BPDUs with the proposal flag set until the forward-delay timer expires. At that time, the port transitions to the forwarding state.

Processing Inferior BPDU Information

An inferior BPDU is a BPDU with root information (such as a higher switch ID or higher path cost) that is inferior to what is currently stored for the port.

If a designated port receives an inferior BPDU, it immediately replies with its own information.

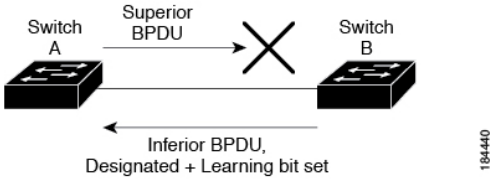
Detecting Unidirectional Link Failure:Rapid PVST+

The software checks the consistency of the port role and state in the received BPDUs to detect unidirectional link failures that could cause bridging loops using the Unidirectional Link Detection (UDLD) feature. This feature is based on the dispute mechanism.

See the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*, for information on UDLD.

When a designated port detects a conflict, it keeps its role, but reverts to a discarding state because disrupting connectivity in case of inconsistency is preferable to opening a bridging loop.

Figure 10: Detecting Unidirectional Link Failure



This figure illustrates a unidirectional link failure that typically creates a bridging loop. Switch A is the root bridge, and its BPDUs are lost on the link leading to switch B. The 802.1w-standard BPDUs include the role and state of the sending port. With this information, switch A can detect that switch B does not react to the superior BPDUs that it sends and that switch B is the designated, not root port. As a result, switch A blocks (or keeps blocking) its port, which prevents the bridging loop.

Port Cost

Note Rapid PVST+ uses the short (16-bit) path-cost method to calculate the cost by default. With the short path-cost method, you can assign any value in the range of 1 to 65535. However, you can configure the device to use the long (32-bit) path-cost method, which allows you to assign any value in the range of 1 to 200,000,000. You configure the path-cost calculation method globally.

This table shows how the STP port path-cost default value is determined from the media speed and path-cost calculation method of a LAN interface.

Table 10: Default Port Cost

Bandwidth	Short Path-Cost Method of Port Cost	Long Path-Cost Method of Port Cost
10 Mbps	100	2,000,000
100 Mbps	19	200,000
1 Gbps	4	20,000
10 Gbps	2	2,000
40 Gbps	1	500
100 Gbps	1	200
400 Gbps	1	50

If a loop occurs, STP considers the port cost when selecting a LAN interface to put into the forwarding state.

You can assign the lower cost values to LAN interfaces that you want STP to select first and higher cost values to LAN interfaces that you want STP to select last. If all LAN interfaces have the same cost value, STP puts the LAN interface with the lowest LAN interface number in the forwarding state and blocks other LAN interfaces.

On access ports, you assign the port cost by the port. On trunk ports, you assign the port cost by the VLAN; you can configure the same port cost to all the VLANs on a trunk port.

Port Priority

If a redundant path occurs and multiple ports have the same path cost, Rapid PVST+ considers the port priority when selecting which LAN port to put into the forwarding state. You can assign lower priority values to LAN ports that you want Rapid PVST+ to select first and higher priority values to LAN ports that you want Rapid PVST+ to select last.

If all LAN ports have the same priority value, Rapid PVST+ puts the LAN port with the lowest LAN port number in the forwarding state and blocks other LAN ports. The possible priority range is from 0 through 224 (the default is 128), configurable in increments of 32. The device uses the port priority value when the LAN port is configured as an access port and uses the VLAN port priority values when the LAN port is configured as a trunk port.

Rapid PVST+ and IEEE 802.1Q Trunks

The 802.1Q trunks impose some limitations on the STP strategy for a network. In a network of Cisco network devices connected through 802.1Q trunks, the network devices maintain one instance of STP for each VLAN allowed on the trunks. However, non-Cisco 802.1Q network devices maintain only one instance of STP for all VLANs allowed on the trunks, which is the Common Spanning Tree (CST).

When you connect a Cisco network device to a non-Cisco device through an 802.1Q trunk, the Cisco network device combines the STP instance of the 802.1Q VLAN of the trunk with the STP instance of the non-Cisco 802.1Q network device. However, all per-VLAN STP information that is maintained by Cisco network devices is separated by a cloud of non-Cisco 802.1Q network devices. The non-Cisco 802.1Q cloud that separates the Cisco network devices is treated as a single trunk link between the network devices.

For more information on 802.1Q trunks, see the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*.

Rapid PVST+ Interoperation with Legacy 802.1D STP

Rapid PVST+ can interoperate with devices that are running the legacy 802.1D protocol. The device knows that it is interoperating with equipment running 802.1D when it receives a BPDU version 0. The BPDUs for Rapid PVST+ are version 2. If the BPDU received is an 802.1w BPDU version 2 with the proposal flag set, the device sends an agreement message after all of the other ports are synchronized. If the BPDU is an 802.1D BPDU version 0, the device does not set the proposal flag and starts the forward-delay timer for the port. The new root port requires twice the forward-delay time to transition to the forwarding state.

The device interoperates with legacy 802.1D devices as follows:

- Notification—Unlike 802.1D BPDUs, 802.1w does not use TCN BPDUs. However, for interoperability with 802.1D devices, the device processes and generates TCN BPDUs.
- Acknowledgment—When an 802.1w device receives a TCN message on a designated port from an 802.1D device, it replies with an 802.1D configuration BPDU with the TCA bit set. However, if the TC-while timer (the same as the TC timer in 802.1D) is active on a root port connected to an 802.1D device and a configuration BPDU with the TCA set is received, the TC-while timer is reset.

  This method of operation is required only for 802.1D devices. The 802.1w BPDUs do not have the TCA bit set.
- Protocol migration—For backward compatibility with 802.1D devices, 802.1w selectively sends 802.1D configuration BPDUs and TCN BPDUs on a per-port basis.

  When a port is initialized, the migrate-delay timer is started (specifies the minimum time during which 802.1w BPDUs are sent), and 802.1w BPDUs are sent. While this timer is active, the device processes all BPDUs received on that port and ignores the protocol type.

  If the device receives an 802.1D BPDU after the port migration-delay timer has expired, it assumes that it is connected to an 802.1D device and starts using only 802.1D BPDUs. However, if the 802.1w device is using 802.1D BPDUs on a port and receives an 802.1w BPDU after the timer has expired, it restarts the timer and starts using 802.1w BPDUs on that port.

Note If you want all devices on the same LAN segment to reinitialize the protocol on each interface, you must reinitialize Rapid PVST+.

Rapid PVST+ Interoperation with 802.1s MST

Rapid PVST+ interoperates seamlessly with the IEEE 802.1s Multiple Spanning Tree (MST) standard. No user configuration is needed. To disable this seamless interoperation, you can use PVST Simulation.

High Availability for Rapid PVST+

The software supports high availability for Rapid PVST+. However, the statistics and timers are not restored when Rapid PVST+ restarts. The timers start again and the statistics begin from 0.

Note See the *Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide*, for complete information on high-availability features.

Prerequisites for Configuring Rapid PVST+

Rapid PVST+ has the following prerequisites:

- You must be logged onto the device.

Guidelines and Limitations for Configuring Rapid PVST+

Rapid PVST+ has the following configuration guidelines and limitations:

- **show** commands with the **internal** keyword are not supported.
- For VLAN configuration limits please see the *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*.
- Port channeling—The port-channel bundle is considered as a single port. The port cost is the aggregation of all the configured port costs assigned to that channel.
- We recommend that you configure all ports connected to Layer 2 hosts as STP edge ports.
- Always leave STP enabled.
- Do not change timers because changing timers can adversely affect stability.
- Keep user traffic off the management VLAN; keep the management VLAN separate from the user data.
- Choose the distribution and core layers as the location of the primary and secondary root switches.
- When you connect two Cisco devices through 802.1Q trunks, the switches exchange spanning tree BPDUs on each VLAN allowed on the trunks. The BPDUs on the native VLAN of the trunk are sent untagged to the reserved 802.1D spanning tree multicast MAC address (01-80-C2-00-00-00). The BPDUs on all VLANs on the trunk are sent tagged to the reserved Cisco Shared Spanning Tree Protocol (SSTP) multicast MAC address (01-00-0c-cc-cc-cd).
- Accurate functionality and visibility of L2 Gateway STP (L2GSTP) are dependent on enabling the spanning-tree domain and assigning a valid domain ID. Without these configurations, L2GSTP may incorrectly appear as disabled in the CLI summary output. After configuring, verify the status using show

spanning-tree summary. The expected output should reflect "L2 Gateway Domain ID: <domain-id>" indicating active functionality.

Guidelines and Limitations on N9300-Series Smart Switches

Beginning with Cisco NX-OS Release 10.6(2)F, you can configure Rapid PVST+ on the Cisco Nexus N9324C-SE1U and N9348Y2C6D-SE1U switches.

Default Settings for Rapid PVST+

This table lists the default settings for Rapid PVST+ parameters.

Table 11: Default Rapid PVST+ Parameters

Parameters	Default
Spanning Tree	Enabled on all VLANs.
Spanning Tree mode	Rapid PVST+ **Caution** Changing the spanning tree mode disrupts the traffic because all spanning tree instances are stopped for the previous mode and started for the new mode.
VLAN	All ports assigned to VLAN1.
Extended system ID	Always enabled.
MAC address reduction	Always enabled.
Bridge ID priority	32769 (default bridge priority plus system ID extension of default VLAN1).
Port state	Blocking (changes immediately after convergence).
Port role	Designated (changes after convergence).
Port/VLAN priority	128.
Path-cost calculation method	Short.

Parameters	Default

Port/VLAN cost	Auto
	The default port cost is determined by the media speed and path-cost method calculation, as follows:
	• 1 Gigabit Ethernet:
	  • short: 4
	  • long: 20,000
	• 10 Gigabit Ethernet:
	  • short: 2
	  • long: 2,000
	• 40 Gigabit Ethernet:
	  • short: 1
	  • long: 500

Hello time	2 seconds.
Forward delay time	15 seconds.
Maximum aging time	20 seconds.
Link type	Auto The default link type is determined by the duplex, as follows: • Full duplex: point-to-point link • Half duplex: shared link

Configuring Rapid PVST+

Rapid PVST+, which has the 802.1 w standard applied to the PVST+ protocol, is the default STP setting in the device.

You enable Rapid PVST+ on a per-VLAN basis. The device maintains a separate instance of STP for each VLAN (except on those VLANS on which you disable STP). By default, Rapid PVST+ is enabled on the default VLAN and on each VLAN that you create.

Enabling Rapid PVST+ - CLI Version

If you disable Rapid PVST+ on any VLANs, you must reenable Rapid PVRST+ on the specified VLANs. If you have enabled MST on the device and now want to use Rapid PVST+, you must enable Rapid PVST+ on the device.

Rapid PVST+ is the default STP mode. You cannot simultaneously run MST and Rapid PVST+ in the same chassis.

Note When you change the spanning tree mode, traffic is disrupted because all spanning tree instances are stopped for the previous mode and started for the new mode.

SUMMARY STEPS

1. **config t**
2. **spanning-tree mode rapid-pvst**
3. **exit**
4. (Optional) **show running-config spanning-tree all**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**spanning-tree mode rapid-pvst** **Example:** `switch(config)# spanning-tree mode rapid-pvst`	Enables Rapid PVST+ on the device. Rapid PVST+ is the default spanning tree mode. **Note** Changing the spanning tree mode disrupts traffic because all spanning tree instances are stopped for the previous mode and started for the new mode.

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

Step 4	(Optional) **show running-config spanning-tree all** **Example:** `switch# show running-config spanning-tree all`	Displays information about the currently running STP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to enable Rapid PVST+ on the device:

```
switch# config t
switch(config)# spanning-tree mode rapid-pvst
switch(config)# exit
switch#
```

Note Because Rapid PVST+ is enabled by default, entering the **show running** command to view the resulting configuration does not display the command that you entered to enable Rapid PVST+.

Disabling or Enabling Rapid PVST+ Per VLAN - CLI Version

You can enable or disable Rapid PVST+ on each VLAN.

Note Rapid PVST+ is enabled by default on the default VLAN and on all VLANs that you create.

SUMMARY STEPS

1. **config t**
2. **spanning-tree vlan** *vlan-range* or **no spanning-tree vlan** *vlan-range*
3. **exit**
4. (Optional) **show spanning-tree**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**spanning-tree vlan** *vlan-range* or **no spanning-tree vlan** *vlan-range* **Example:** `switch(config)# spanning-tree vlan 5`	• **spanning-tree vlan** *vlan-range* Enables Rapid PVST+ (default STP) on a per VLAN basis. The *vlan-range* value can be 2 through 3967 except for reserved VLAN values. • **no spanning-tree vlan** *vlan-range*

	Command or Action	Purpose
		Disables Rapid PVST+ on the specified VLAN. See the Caution for information regarding this command.

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

Step 4	(Optional) **show spanning-tree** **Example:** `switch# show spanning-tree`	Displays the STP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to enable STP on VLAN 5:

```
switch# config t
switch(config)# spanning-tree vlan 5
switch(config)# exit
switch#
```

Note Do not disable spanning tree on a VLAN unless all switches and bridges in the VLAN have spanning tree disabled. You cannot disable spanning tree on some switches and bridges in a VLAN and leave it enabled on other switches and bridges in the VLAN. This action can have unexpected results because switches and bridges with spanning tree enabled will have incomplete information regarding the physical topology of the network.

Caution We do not recommend disabling spanning tree even in a topology that is free of physical loops. Spanning tree serves as a safeguard against misconfigurations and cabling errors. Do not disable spanning tree in a VLAN without ensuring that no physical loops are present in the VLAN.

Note Because STP is enabled by default, entering the **show running** command to view the resulting configuration does not display the command that you entered to enable STP.

Configuring the Root Bridge ID

The device maintains a separate instance of STP for each active VLAN in Rapid PVST+. For each VLAN, the network device with the lowest bridge ID becomes the root bridge for that VLAN.

To configure a VLAN instance to become the root bridge, modify the bridge priority from the default value (32768) to a significantly lower value.

When you enter the **spanning-tree vlan** *vlan-range* **root primary** command, the device sets the bridge priority to 24576 if this value will cause the device to become the root for the specified VLANs. If any root bridge for the specified VLAN has a bridge priority lower than 24576, the device sets the bridge priority for the specified VLANs to 4096 less than the lowest bridge priority.

Caution The root bridge for each instance of STP should be a backbone or distribution device. Do not configure an access device as the STP primary root.

Note With the device configured as the root bridge, do not manually configure the hello time, forward-delay time, and maximum-age time using the **spanning-tree mst hello-time**, **spanning-tree mst forward-time**, and **spanning-tree mst max-age** global configuration commands.

SUMMARY STEPS

1. **config t**
2. **spanning-tree vlan** *vlan-range* **root primary**
3. **exit**
4. (Optional) **show spanning-tree**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**spanning-tree vlan** *vlan-range* **root primary** **Example:** `switch(config)# spanning-tree vlan 2 root primary`	Sets the bridge priority for the spanning tree.
Step 3	**exit** **Example:**	Exits configuration mode.

	Command or Action	Purpose
	`switch(config)# exit` `switch#`	
Step 4	(Optional) **show spanning-tree** **Example:** `switch# show spanning-tree`	Displays the STP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the device as the root bridge:

```
switch# config t
switch(config)# spanning-tree vlan 2 root primary
switch(config)# exit
switch#
```

Configuring a Secondary Root Bridge-CLI Version

When you configure a device as the secondary root, the STP bridge priority is modified from the default value (32768) so that the device is likely to become the root bridge for the specified VLANs if the primary root bridge fails (assuming the other network devices in the network use the default bridge priority of 32768). STP sets the bridge priority to 28672.

Enter the **diameter** keyword to specify the Layer 2 network diameter (that is, the maximum number of bridge hops between any two end stations in the Layer 2 network). When you specify the network diameter, the software automatically selects an optimal hello time, forward delay time, and maximum age time for a network of that diameter, which can significantly reduce the STP convergence time. You can enter the **hello-time** keyword to override the automatically calculated hello time.

You can configure more than one device in this manner to have multiple backup root bridges. Enter the same network diameter and hello time values that you used when configuring the primary root bridge.

Note With the device configured as the root bridge, do not manually configure the hello time, forward-delay time, and maximum-age time using the **spanning-tree mst hello-time**, **spanning-tree mst forward-time**, and **spanning-tree mst max-age** global configuration commands.

SUMMARY STEPS

1. **config t**
2. **spanning-tree vlan** *vlan-range* **root secondary** [**diameter** *dia* [**hello-time** *hello-time*]]
3. **exit**
4. (Optional) **show spanning-tree vlan** *vlan_id*

5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**spanning-tree vlan** *vlan-range* **root secondary** [**diameter** *dia* [**hello-time** *hello-time*]] **Example:** `switch(config)# spanning-tree vlan 5 root secondary diameter 4`	Configures a device as the secondary root bridge. The *vlan-range* value can be 2 through 3967 (except for reserved VLAN values). The *dia* default is 7. The *hello-time* can be from 1 to 10 seconds, and the default value is 2 seconds.

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

Step 4	(Optional) **show spanning-tree vlan** *vlan_id* **Example:** `switch# show spanning-tree vlan 5`	Displays the STP configuration for the specified VLANs.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the device as the secondary root bridge for VLAN 5 with a network diameter of 4:

```
switch# config t
switch(config)# spanning-tree vlan 5 root secondary diameter 4
switch(config)# exit
switch#
```

Configuring the Rapid PVST+ Bridge Priority of a VLAN

You can configure the Rapid PVST+ bridge priority of a VLAN. This is another method of configuring root bridges.

Note Be careful when using this configuration. We recommend that you configure the primary root and secondary root to modify the bridge priority.

SUMMARY STEPS

1. **config t**
2. **spanning-tree vlan** *vlan-range* **priority** *value*
3. **exit**
4. (Optional) **show spanning-tree vlan** *vlan_id*
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**spanning-tree vlan** *vlan-range* **priority** *value* **Example:** `switch(config)# spanning-tree vlan 5 priority 8192`	Configures the bridge priority of a VLAN. Valid values are 0, 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768, 36864, 40960, 45056, 49152, 53248, 57344, and 61440. All other values are rejected. The default value is 32768.

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

Step 4	(Optional) **show spanning-tree vlan** *vlan_id* **Example:** `switch# show spanning-tree vlan 5`	Displays the STP configuration for the specified VLANs.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the priority of VLAN 5 on Gigabit Ethernet port 1/4 to 8192:

```
switch# config t
switch(config)# spanning-tree vlan 5 priority 8192
```

```
switch(config)# exit
switch#
```

Configuring the Rapid PVST+ Port Priority - CLI Version

You can assign lower priority values to LAN ports that you want Rapid PVST+ to select first and higher priority values to LAN ports that you want Rapid PVST+ to select last. If all LAN ports have the same priority value, Rapid PVST+ puts the LAN port with the lowest LAN port number in the forwarding state and blocks other LAN ports.

The device uses the port priority value when the LAN port is configured as an access port and uses the VLAN port priority values when the LAN port is configured as a trunk port.

SUMMARY STEPS

1. **config t**
2. **interface** *type slot/port*
3. **spanning-tree** [**vlan** *vlan-list*] **port-priority** *priority*
4. **exit**
5. (Optional) **show spanning-tree interface** {**ethernet** *slot/port* | *port channel channel-number*}
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)#`
		Specifies the interface to configure and enters the interface configuration mode.

Step 3	**spanning-tree** [**vlan** *vlan-list*] **port-priority** *priority* **Example:** `switch(config-if)# spanning-tree port-priority 160`	Configures the port priority for the LAN interface. The *priority* value can be from 0 to 224. A lower value indicates a higher priority. The priority values are 0, 32, 64, 96, 128, 160, 192, and 224. All other values are rejected. The default value is 128.

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

	Command or Action	Purpose
Step 5	(Optional) **show spanning-tree interface** {**ethernet** *slot/port* | *port channel channel-number*} **Example:** `switch# show spanning-tree interface ethernet 2/10`	Displays the STP configuration for the specified interface.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the port priority of Ethernet access port 1/4 to 160:

```
switch# config t
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree port-priority 160
switch(config-if)# exit
switch(config)#
```

Configuring the Rapid PVST+ Path-Cost Method and Port Cost - CLI Version

On access ports, you can assign the port cost for each port. On trunk ports, you can assign the port cost for each VLAN; you can configure all the VLANs on a trunk with the same port cost.

Note In Rapid PVST+ mode, you can use either the short or long path-cost method, and you can configure the method in either the interface or configuration submode. The default path-cost method is short.

SUMMARY STEPS

1. **config t**
2. **spanning-tree pathcost method** {**long** | **short**}
3. **interface** *type slot/port*
4. **spanning-tree** [**vlan** *vlan-id*] **cost** [*value* | *auto*]
5. **exit**
6. (Optional) **show spanning-tree pathcost method**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**spanning-tree pathcost method** {**long** | **short**} **Example:** `switch(config)# spanning-tree pathcost method long`	Selects the method used for Rapid PVST+ path-cost calculations. The default method is the short method.

Step 3	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)`
		Specifies the interface to configure and enters the interface configuration mode.

Step 4	**spanning-tree** [**vlan** *vlan-id*] **cost** [*value* | *auto*] **Example:** `switch(config-if)# spanning-tree cost 1000`	Configures the port cost for the LAN interface. The cost value, depending on the path-cost calculation method, can be as follows: • short—1 to 65535 • long—1 to 200000000 **Note** You configure this parameter per port on access ports and per VLAN on trunk ports. The default is **auto**, which sets the port cost on both the path-cost calculation method and the media speed.

Step 5	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface mode.

Step 6	(Optional) **show spanning-tree pathcost method** **Example:** `switch# show spanning-tree pathcost method`	Displays the STP path-cost method.
Step 7	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the port cost of Ethernet access port 1/4 to 1000:

```
switch# config t
switch (config)# spanning-tree pathcost method long
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree cost 1000
switch(config-if)# exit
switch(config)#
```

Configuring the Rapid PVST+ Hello Time for a VLAN - CLI Version

You can configure the Rapid-PVST+ hello time for a VLAN.

Note Be careful when using this configuration because you may disrupt the Spanning Tree. For most situations, we recommend that you configure the primary root and secondary root to modify the hello time.

SUMMARY STEPS

1. **config t**
2. **spanning-tree vlan** *vlan-range* **hello-time** *value*
3. **exit**
4. (Optional) **show spanning-tree vlan** *vlan_id*
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**spanning-tree vlan** *vlan-range* **hello-time** *value* **Example:** `switch(config)# spanning-tree vlan 5 hello-time 7`	Configures the hello time of a VLAN. The hello time value can be from 1 to 10 seconds, and the default is 2 seconds.

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

	Command or Action	Purpose
Step 4	(Optional) **show spanning-tree vlan** *vlan_id* **Example:** `switch# show spanning-tree vlan 5`	Displays the STP configuration per VLAN.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the hello time for VLAN 5 to 7 seconds:

```
switch# config t
switch(config)# spanning-tree vlan 5 hello-time 7
switch(config)# exit
switch#
```

Configuring the Rapid PVST+ Forward Delay Time for a VLAN - CLI Version

You can configure the forward delay time per VLAN when using Rapid PVST+.

SUMMARY STEPS

1. **config t**
2. **spanning-tree vlan** *vlan-range* **forward-time** *value*
3. **exit**
4. (Optional) **show spanning-tree vlan** *vlan_id*
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**spanning-tree vlan** *vlan-range* **forward-time** *value* **Example:** `switch(config)# spanning-tree vlan 5 forward-time 21`	Configures the forward delay time of a VLAN. The forward delay time value can be from 4 to 30 seconds, and the default is 15 seconds.

	Command or Action	Purpose

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

Step 4	(Optional) **show spanning-tree vlan** *vlan_id* **Example:** `switch# show spanning-tree vlan 5`	Displays the STP configuration per VLAN.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the forward delay time for VLAN 5 to 21 seconds:

```
switch# config t
switch(config)# spanning-tree vlan 5 forward-time 21
switch(config)# exit
switch#
```

Configuring the Rapid PVST+ Maximum Age Time for a VLAN - CLI Version

You can configure the maximum age time per VLAN when using Rapid PVST+.

SUMMARY STEPS

1. **config t**
2. **spanning-tree vlan** *vlan-range* **max-age** *value*
3. **exit**
4. (Optional) **show spanning-tree vlan** *vlan_id*
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

	Command or Action	Purpose
Step 2	**spanning-tree vlan** *vlan-range* **max-age** *value* **Example:** `switch(config)# spanning-tree vlan 5 max-age 36`	Configures the maximum aging time of a VLAN. The maximum aging time value can be from 6 to 40 seconds, and the default is 20 seconds.

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

Step 4	(Optional) **show spanning-tree vlan** *vlan_id* **Example:** `switch# show spanning-tree vlan 5`	Displays the STP configuration per VLAN.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the maximum aging time for VLAN 5 to 36 seconds:

```
switch# config t
switch(config)# spanning-tree vlan 5 max-age 36
switch(config)# exit
switch#
```

Specifying the Link Type for Rapid PVST+ - CLI Version

Rapid connectivity (802.1w standard) is established only on point-to-point links. By default, the link type is controlled from the duplex mode of the interface. A full-duplex port is considered to have a point-to-point connection; a half-duplex port is considered to have a shared connection.

If you have a half-duplex link physically connected point to point to a single port on a remote device, you can override the default setting on the link type and enable rapid transitions.

If you set the link to shared, STP falls back to 802.1D.

SUMMARY STEPS

1. **config t**
2. **interface** *type slot/port*
3. **spanning-tree link-type** {*auto* | *point-to-point* | *shared*}
4. **exit**
5. (Optional) **show spanning-tree**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)#`
		Specifies the interface to configure and enters the interface configuration mode.

Step 3	**spanning-tree link-type** {*auto* | *point-to-point* | *shared*} **Example:** `switch(config-if)# spanning-tree link-type point-to-point`	Configures the link type to be either a point-to-point link or shared link. The system reads the default value from the device connection, as follows: half duplex links are shared and full-duplex links are point to point. If the link type is shared, the STP falls back to 802.1D. The default is auto, which sets the link type based on the duplex setting of the interface.

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

Step 5	(Optional) **show spanning-tree** **Example:** `switch# show spanning-tree`	Displays the STP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the link type as a point-to-point link:

```
switch# config t
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree link-type point-to-point
switch(config-if)# exit
switch(config)#
```

Reinitializing the Protocol for Rapid PVST+

A bridge that runs Rapid PVST+ can send 802.1D BPDUs on one of its ports when it is connected to a legacy bridge. However, the STP protocol migration cannot determine whether the legacy device has been removed from the link unless the legacy device is the designated switch. You can reinitialize the protocol negotiation (force the renegotiation with neighboring devices) on the entire device or on specified interfaces.

SUMMARY STEPS

1. **clear spanning-tree detected-protocol** [**interface** {**ethernet** *slot/port* | **port channel** *channel-number*}]

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**clear spanning-tree detected-protocol** [**interface** {**ethernet** *slot/port* | **port channel** *channel-number*}] **Example:** `switch# clear spanning-tree detected-protocol`	Reinitializes Rapid PVST+ on all interfaces on the device or specified interfaces.

Example

This example shows how to reinitialize Rapid PVST+ on the Ethernet interface on slot 2, port 8:

```
switch# clear spanning-tree detected-protocol interface ethernet 2/8
switch#
```

Verifying the Rapid PVST+ Configurations

To display Rapid PVST+ configuration information, perform one of the following tasks:

Command	Purpose
show running-config spanning-tree [**all**]	Displays STP information.
show spanning-tree summary	Displays summary STP information.
show spanning-tree detail	Displays detailed STP information.
show spanning-treeshow spanning-tree {**vlan***vlan-id* | **interface** {[**ethernet***slot/port*] | [**port-channel***channel-number*]}} [**detail**]	Displays STP information per VLAN and interface.
show spanning-tree vlanshow spanning-tree vlan *vlan-id* **bridge**	Displays information on the STP bridge.

Displaying and Clearing Rapid PVST+ Statistics -- CLI Version

To display Rapid PVRST+ configuration information, perform one of the following tasks:

Command	Purpose
clear spanning-tree counters [**interface** *type slot/port* | **vlan***vlan-id*]	Clears the counters for STP.
show spanning-tree {**vlan** *vlan-id* | **interface** {[**ethernet** *slot/port*] | [**port-channel** *channel-number*]}} **detail**	Displays information about STP by interface or VLAN including BPDUs sent and received.

Rapid PVST+ Example Configurations

The following example shows how to configure Rapid PVST+:

```
switch# configure terminal
switch(config)# spanning-tree port type edge bpduguard default
switch(config)# spanning-tree port type edge bpdufilter default
switch(config)# spanning-tree port type network default
switch(config)# spanning-tree vlan 1-10 priority 24576
switch(config)# spanning-tree vlan 1-10 hello-time 1
switch(config)# spanning-tree vlan 1-10 forward-time 9
switch(config)# spanning-tree vlan 1-10 max-age 13

switch(config)# interface Ethernet 3/1 switchport
switch(config-if)# spanning-tree port type edge
switch(config-if)# exit

switch(config)# spanning-tree port type edge
switch(config-if)# switchport
switch(config-if)# switchport mode trunk
switch(config-if)# spanning-tree guard root
switch(config-if)# exit
switch(config)#
```

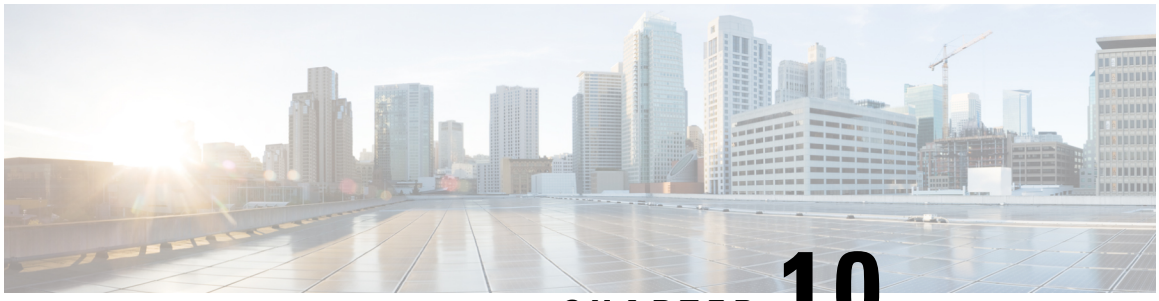
Additional References for Rapid PVST+ -- CLI Version

Related Documents

Related Topic	Document Title
Layer 2 interfaces	*Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*
Cisco NX-OS fundamentals	*Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide*
System management	*Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*

Standards

Standards	Title
IEEE 802.1Q-2006 (formerly known as IEEE 802.1s), IEEE 802.1D-2004 (formerly known as IEEE 802.1w), IEEE 802.1D, IEEE 802.1t	—



CHAPTER 10

Configuring MST Using Cisco NX-OS

- Information About MST, on page 135
- Prerequisites for MST, on page 143
- Guidelines and Limitations for Configuring MST, on page 143
- Default Settings for MST, on page 144
- Configuring MST, on page 145
- Verifying the MST Configuration, on page 171
- Displaying and Clearing MST Statistics -- CLI Version, on page 171
- MST Example Configuration, on page 172
- Additional References for MST -- CLI Version, on page 173

Information About MST

Note See the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*, for information on creating Layer 2 interfaces.

MST, which is the IEEE 802.1s standard, allows you to assign two or more VLANs to a spanning tree instance. MST is not the default spanning tree mode; Rapid per VLAN Spanning Tree (Rapid PVST+) is the default mode. MST instances with the same name, revision number, and VLAN-to-instance mapping combine to form an MST region. The MST region appears as a single bridge to spanning tree configurations outside the region. MST forms a boundary to that interface when it receives an IEEE 802.1D Spanning Tree Protocol (STP) message from a neighboring device.

Note Spanning tree is used to refer to IEEE 802.1w and IEEE 802.1s. If the IEEE 802.1D Spanning Tree Protocol is discussed in this publication, 802.1D is stated specifically.

MST Overview

Note You must enable MST; Rapid PVST+ is the default spanning tree mode.

MST maps multiple VLANs into a spanning tree instance, with each instance having a spanning tree topology independent of other spanning tree instances. This architecture provides multiple forwarding paths for data traffic, enables load balancing, and reduces the number of STP instances required to support a large number of VLANs. MST improves the fault tolerance of the network because a failure in one instance (forwarding path) does not affect other instances (forwarding paths).

MST provides rapid convergence through explicit handshaking because each MST instance uses the IEEE 802.1w standard, which eliminates the 802.1D forwarding delay and quickly transitions root bridge ports and designated ports to the forwarding state.

MAC address reduction is always enabled on the device. You cannot disable this feature.

MST improves spanning tree operation and maintains backward compatibility with these STP versions:

- Original 802.1D spanning tree
- Rapid per-VLAN spanning tree (Rapid PVST+)

Note

- IEEE 802.1 was defined in the Rapid Spanning Tree Protocol (RSTP) and was incorporated into IEEE 802.1D.
- IEEE 802.1 was defined in MST and was incorporated into IEEE 802.1Q

  .

MST Regions

To allow devices to participate in MST instances, you must consistently configure the devices with the same MST configuration information.

A collection of interconnected devices that have the same MST configuration is an MST region. An MST region is a linked group of MST bridges with the same MST configuration.

The MST configuration controls the MST region to which each device belongs. The configuration includes the name of the region, the revision number, and the VLAN-to-MST instance assignment mapping.

A region can have one or multiple members with the same MST configuration. Each member must be capable of processing 802.1w bridge protocol data units (BPDUs). There is no limit to the number of MST regions in a network.

Each device can support up to 65 MST instances (MSTIs), including Instance 0, in a single MST region. Instances are identified by any number in the range from 1 to 4094. The system reserves Instance 0 for a special instance, which is the IST. You can assign a VLAN to only one MST instance at a time.

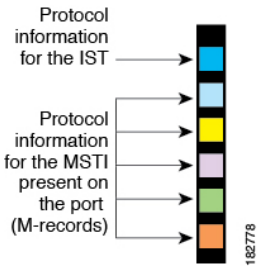
The MST region appears as a single bridge to adjacent MST regions and to other Rapid PVST+ regions and 802.1D spanning tree protocols.

Note We do not recommend that you partition the network into a large number of regions.

MST BPDUs

Each device has only one MST BPDU per interface, and that BPDU carries an M-record for each MSTI on the device. Only the IST sends BPDUs for the MST region; all M-records are encapsulated in that one BPDU that the IST sends. Because the MST BPDU carries information for all instances, the number of BPDUs that need to be processed to support MST is significantly reduced compared with Rapid PVST+.

Figure 11: MST BPDU with M-Records for MSTIs



MST Configuration Information

The MST configuration that must be identical on all devices within a single MST region is configured by the user.

You can configure the three parameters of the MST configuration as follows:

- Name—32-character string, null padded and null terminated, identifying the MST region
- Revision number—Unsigned 16-bit number that identifies the revision of the current MST configuration

Note You must set the revision number when required as part of the MST configuration. The revision number is not incremented automatically each time that the MST configuration is committed.

- VLAN-to-MST instance mapping—4096-element table that associates each of the potential VLANs supported to a given instance with the first (0) and last element (4095) set to 0. The value of element number X represents the instance to which VLAN X is mapped.

Note When you change the VLAN-to-MSTI mapping, the system reconverges MST.

MST BPDUs contain these three configuration parameters. An MST bridge accepts an MST BPDU into its own region only if these three configuration parameters match exactly. If one configuration attribute differs, the MST bridge considers the BPDU to be from another MST region.

IST, CIST, and CST

IST, CIST, and CST Overview

Unlike Rapid PVST+, in which all the STP instances are independent, MST establishes and maintains IST, CIST, and CST spanning trees, as follows:

- An IST is the spanning tree that runs in an MST region.

  MST establishes and maintains additional spanning trees within each MST region; these spanning trees are called multiple spanning tree instances (MSTIs).

  Instance 0 is a special instance for a region, known as the IST. The IST always exists on all ports; you cannot delete the IST, or Instance 0. By default, all VLANs are assigned to the IST. All other MST instances are numbered from 1 to 4094.

  The IST is the only STP instance that sends and receives BPDUs. All of the other MSTI information is contained in MST records (M-records), which are encapsulated within MST BPDUs.

  All MSTIs within the same region share the same protocol timers, but each MSTI has its own topology parameters, such as the root bridge ID, the root path cost, and so forth.

  An MSTI is local to the region; for example, MSTI 9 in region A is independent of MSTI 9 in region B, even if regions A and B are interconnected. Only CST information crosses region boundaries.

- The CST interconnects the MST regions and any instance of 802.1D and 802.1w STP that may be running on the network. The CST is the one STP instance for the entire bridged network and encompasses all MST regions and 802.1w and 802.1D instances.

- A CIST is a collection of the ISTs in each MST region. The CIST is the same as an IST inside an MST region, and the same as a CST outside an MST region.

The spanning tree computed in an MST region appears as a subtree in the CST that encompasses the entire switched domain. The CIST is formed by the spanning tree algorithm running among devices that support the 802.1w, 802.1s, and 802.1D standards. The CIST inside an MST region is the same as the CST outside a region.

Spanning Tree Operation Within an MST Region

The IST connects all the MSTdevices in a region. When the IST converges, the root of the IST becomes the CIST regional root. The CIST regional root is also the CIST root if there is only one region in the network. If the CIST root is outside the region, the protocol selects one of the MST devices at the boundary of the region as the CIST regional root.

When an MST device initializes, it sends BPDUs that identify itself as the root of the CIST and the CIST regional root, with both the path costs to the CIST root and to the CIST regional root set to zero. The device also initializes all of its MSTIs and claims to be the root for all of them. If the device receives superior MSTI root information (lower switch ID, lower path cost, and so forth) than the information that is currently stored for the port, it relinquishes its claim as the CIST regional root.

During initialization, an MST region might have many subregions, each with its own CIST regional root. As devices receive superior IST information from a neighbor in the same region, they leave their old subregions and join the new subregion that contains the true CIST regional root. This action causes all subregions to shrink except for the subregion that contains the true CIST regional root.

All devices in the MST region must agree on the same CIST regional root. Any two devices in the region will only synchronize their port roles for an MSTI if they converge to a common CIST regional root.

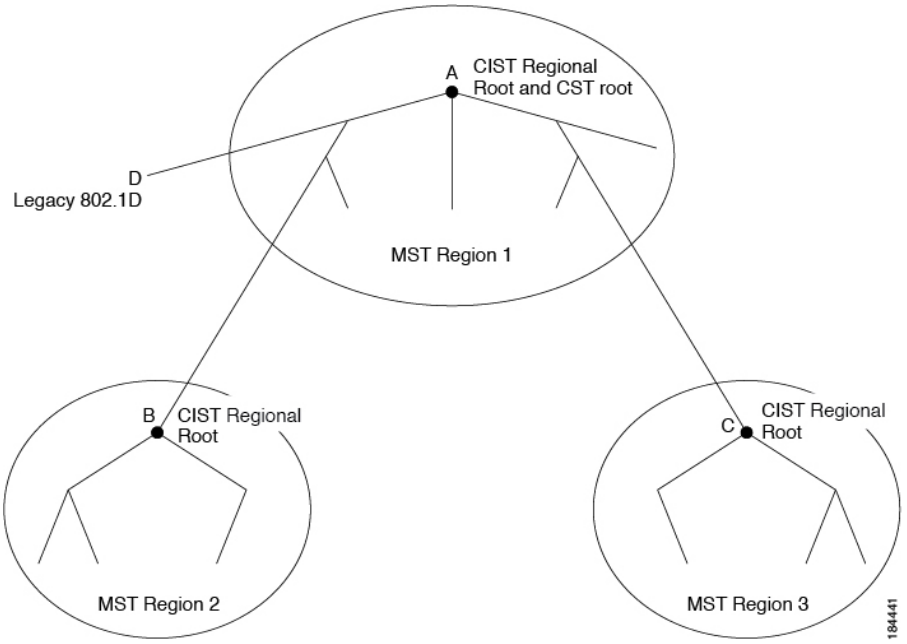
Spanning Tree Operations Between MST Regions

If you have multiple regions or 802.1 w or 802.1D STP instances within a network, MST establishes and maintains the CST, which includes all MST regions and all 802.1w and 802.1D STP devices in the network. The MSTIs combine with the IST at the boundary of the region to become the CST.

The IST connects all the MST devices in the region and appears as a subtree in the CIST that encompasses the entire switched domain. The root of the subtree is the CIST regional root. The MST region appears as a virtual device to adjacent STP devices and MST regions.

Figure 12: MST Regions, CIST Regional Roots, and CST Root

This figure shows a network with three MST regions and an 802.1D device (D). The CIST regional root for region 1 (A) is also the CIST root. The CIST regional root for region 2 (B) and the CIST regional root for region 3 (C) are the roots for their respective subtrees within the CIST.



Only the CST instance sends and receives BPDUs. MSTIs add their spanning tree information into the BPDUs (as M-records) to interact with neighboring devices within the same MST region and compute the final spanning tree topology. The spanning tree parameters related to the BPDU transmission (for example, hello time, forward time, max-age, and max-hops) are configured only on the CST instance but affect all MSTIs. You can configure the parameters related to the spanning tree topology (for example, the switch priority, the port VLAN cost, and the port VLAN priority) on both the CST instance and the MSTI.

MST devices use Version 3 BPDUs. If the MST device falls back to 802.1D STP, the device uses only 802.1D BPDUs to communicate with 802.1D-only devices. MST devices use MST BPDUs to communicate with MST devices.

MST Terminology

MST naming conventions include identification of some internal or regional parameters. These parameters are used only within an MST region, compared to external parameters that are used throughout the whole

network. Because the CIST is the only spanning tree instance that spans the whole network, only the CIST parameters require the external qualifiers and not the internal or regional qualifiers. The MST terminology is as follows:

- The CIST root is the root bridge for the CIST, which is the unique instance that spans the whole network.
- The CIST external root path cost is the cost to the CIST root. This cost is left unchanged within an MST region. An MST region looks like a single device to the CIST. The CIST external root path cost is the root path cost calculated between these virtual devices and devices that do not belong to any region.
- If the CIST root is in the region, the CIST regional root is the CIST root. Otherwise, the CIST regional root is the closest device to the CIST root in the region. The CIST regional root acts as a root bridge for the IST.
- The CIST internal root path cost is the cost to the CIST regional root in a region. This cost is only relevant to the IST, instance 0.

Hop Count

MST does not use the message-age and maximum-age information in the configuration BPDU to compute the STP topology inside the MST region. Instead, the protocol uses the path cost to the root and a hop-count mechanism similar to the IP time-to-live (TTL) mechanism.

By using the **spanning-tree mst max-hops** global configuration command, you can configure the maximum hops inside the region and apply it to the IST and all MST instances in that region.

The hop count achieves the same result as the message-age information (triggers a reconfiguration). The root bridge of the instance always sends a BPDU (or M-record) with a cost of 0 and the hop count set to the maximum value. When a device receives this BPDU, it decrements the received remaining hop count by one and propagates this value as the remaining hop count in the BPDUs that it generates. When the count reaches zero, the device discards the BPDU and ages the information held for the port.

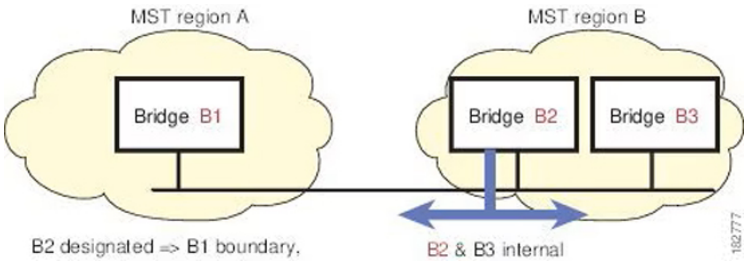
The message-age and maximum-age information in the 802.1w portion of the BPDU remain the same throughout the region (only on the IST), and the same values are propagated by the region-designated ports at the boundary.

You configure a maximum aging time as the number of seconds that a device waits without receiving spanning tree configuration messages before attempting a reconfiguration.

Boundary Ports

A boundary port is a port that connects to a LAN, the designated bridge of which is either a bridge with a different MST configuration (and so, a separate MST region) or a Rapid PVST+ or 802.1D STP bridge. A designated port knows that it is on the boundary if it detects an STP bridge or receives an agreement proposal from an MST bridge with a different configuration or a Rapid PVST+ bridge. This definition allows two ports that are internal to a region to share a segment with a port that belongs to a different region, creating the possibility of receiving both internal and external messages on a port.

Figure 13: MST Boundary Ports



At the boundary, the roles of MST ports do not matter; the system forces their state to be the same as the IST port state. If the boundary flag is set for the port, the MST port-role selection process assigns a port role to the boundary and assigns the same state as the state of the IST port. The IST port at the boundary can take up any port role except a backup port role.

Detecting Unidirectional Link Failure: MST

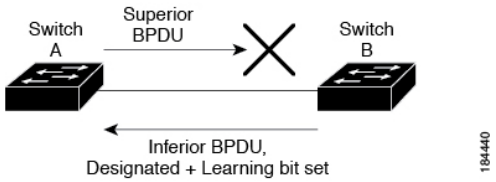
Currently, this feature is not present in the IEEE MST standard, but it is included in the standard-compliant implementation; it is based on the dispute mechanism. The software checks the consistency of the port role and state in the received BPDUs to detect unidirectional link failures that could cause bridging loops. This feature is based on the dispute mechanism.

Note See the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*, for information on Unidirectional Link Detection (UDLD).

When a designated port detects a conflict, it keeps its role, but reverts to a discarding state because disrupting connectivity in case of inconsistency is preferable to opening a bridging loop.

Figure 14: Detecting a Unidirectional Link Failure

This figure shows a unidirectional link failure that typically creates a bridging loop. Switch A is the root bridge, and its BPDUs are lost on the link leading to switch B. Rapid PVST+ (802.1w) and MST BPDUs include the role and state of the sending port. With this information, switch A can detect that switch B does not react to the superior BPDUs that it sends and that switch B is the designated, not root port. As a result, switch A blocks (or keeps blocking) its port, which prevents the bridging loop.



Port Cost and Port Priority

Spanning tree uses port costs to break a tie for the designated port. Lower values indicate lower port costs, and spanning tree chooses the least costly path. Default port costs are taken from the bandwidth of the interface, as follows:

- 1 Gigabit Ethernet—20,000
- 10 Gigabit Ethernet—2,000
- 40 Gigabit Ethernet—500

You can configure the port costs in order to influence which port is chosen.

Note MST always uses the long path-cost calculation method, so the range of valid values is between 1 and 200,000,000.

The system uses port priorities to break ties among ports with the same cost. A lower number indicates a higher priority. The default port priority is 128. You can configure the priority to values between 0 and 224, in increments of 32.

Interoperability with IEEE 802.1D

A device that runs MST supports a built-in protocol migration feature that enables it to interoperate with 802.1D STP devices. If this device receives an 802.1D configuration BPDU (a BPDU with the protocol version set to 0), it sends only 802.1D BPDUs on that port. In addition, an MST device can detect that a port is at the boundary of a region when it receives an 802.1D BPDU, an MST BPDU (Version 3) associated with a different region, or an 802.1w BPDU (Version 2).

However, the device does not automatically revert to the MST mode if it no longer receives 802.1D BPDUs because it cannot detect whether the 802.1D device has been removed from the link unless the 802.1D device is the designated device. A device might also continue to assign a boundary role to a port when the device to which this device is connected has joined the region.

To restart the protocol migration process (force the renegotiation with neighboring devices), enter the **clear spanning-tree detected-protocols** command.

All Rapid PVST+ switches (and all 8021.D STP switches) on the link can process MST BPDUs as if they are 802.1w BPDUs. MST devices can send either Version 0 configuration and topology change notification (TCN) BPDUs or Version 3 MST BPDUs on a boundary port. A boundary port connects to a LAN, the designated device of which is either a single spanning tree device or a device with a different MST configuration.

MST interoperates with the Cisco prestandard MSTP whenever it receives prestandard MSTP on an MST port; no explicit configuration is necessary.

You can also configure the interface to proactively send prestandard MSTP messages.

High Availability for MST

The software supports high availability for MST. However, the statistics and timers are not restored when MST restarts. The timers start again and the statistics begin from 0.

The device supports full nondisruptive upgrades for MST. See the *Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide*, for complete information on nondisruptive upgrades and high-availability features.

Prerequisites for MST

MST has the following prerequisites:

- You must be logged onto the device.

Guidelines and Limitations for Configuring MST

Note When you change the VLAN-to-MSTI mapping, the system reconverges MST.

MST has the following configuration guidelines and limitations:

- For MST configuration limits, see the *Cisco Nexus 9000 Series NX-OS Verified Scalability Guide*.
- **show** commands with the **internal** keyword are not supported.
- You must enable MST; Rapid PVST+ is the default spanning tree mode.
- You can assign a VLAN to only one MST instance at a time.
- You cannot map VLANs 3968 to 4095 to an MST instance. These VLANs are reserved for internal use by the device.
- You can have up to 65 MST instances on one device.
- By default, all VLANs are mapped to MSTI 0 or the IST.
- You can load balance only within the MST region.
- Ensure that trunks within an MST region carry all of the VLANs that are mapped to an MSTI or exclude all those VLANs that are mapped to an MSTI.
- Always leave STP enabled.
- Do not change timers because you can adversely affect your network stability.
- Keep user traffic off the management VLAN; keep the management VLAN separate from user data.
- Choose the distribution and core layers as the location of the primary and secondary root switches.
- Port channeling—The port channel bundle is considered as a single port. The port cost is the aggregation of all the configured port costs assigned to that channel.
- When you map a VLAN to an MSTI, the system automatically removes that VLAN from its previous MSTI.
- You can map any number of VLANs to an MSTI.
- All MST boundary ports must be forwarding for load balancing between Rapid PVST+ and an MST cloud or between a PVST+ and an MST cloud. The CIST regional root of the MST cloud must be the root of the CST. If the MST cloud consists of multiple MST regions, one of the MST regions must contain

the CST root and all of the other MST regions must have a better path to the root contained within the MST cloud than a path through the Rapid PVST+ or PVST+ cloud.

- Do not partition the network into a large number of regions. However, if this situation is unavoidable, we recommend that you partition the switched LAN into smaller LANs interconnected by non-Layer 2 devices.
- When you are in the MST configuration submode, the following guidelines apply:
  - Each command reference line creates its pending regional configuration.
  - The pending region configuration starts with the current region configuration.
  - To leave the MST configuration submode without committing any changes, enter the **abort** command.
  - To leave the MST configuration submode and commit all the changes that you made before you left the submode, enter the **exit** or **end** commands, or press **Ctrl + Z**.

Note The software supports full nondisruptive upgrades for MST.

Guidelines and Limitations on N9300-Series Smart Switches

Beginning with Cisco NX-OS Release 10.6(2)F, you can configure Multiple Spanning Tree (MST) on the Cisco Nexus N9324C-SE1U and N9348Y2C6D-SE1U switches.

Default Settings for MST

This table lists the default settings for MST parameters.

Table 12: Default MST Parameters

Parameters	Default
Spanning tree	Enabled
Spanning tree mode	Rapid PVST+ is enabled by default **Caution** Changing the spanning tree mode disrupts the traffic because all spanning tree instances are stopped for the previous mode and started for the new mode.
Name	Empty string
VLAN mapping	All VLANs mapped to a CIST instance
Revision	0
Instance ID	Instance 0; VLANs 1 to 3967 are mapped to Instance 0 by default

Parameters	Default
MSTIs per MST region	65
Bridge priority (configurable per CIST port)	32768
Spanning tree port priority (configurable per CIST port)	128
Spanning tree port cost (configurable per CIST port)	Auto The default port cost is determined by the port speed as follows: • 1 Gigabit Ethernet: 20,000 • 10 Gigabit Ethernet: 2,000 • 40 Gigabit Ethernet: 500
Hello time	2 seconds
Forward-delay time	15 seconds
Maximum-aging time	20 seconds
Maximum hop count	20 hops
Link type	Auto The default link type is determined by the duplex, as follows: • Full duplex: point-to-point link • Half duplex: shared link

Configuring MST

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco software commands for this feature might differ from the Cisco IOS commands that you would use.

Enabling MST - CLI Version

You can enable MST; Rapid PVST+ is the default.

Note When you change the spanning tree mode, traffic is disrupted because all spanning tree instances are stopped for the previous mode and started for the new mode.

SUMMARY STEPS

1. **config t**
2. **spanning-tree mode mst** or **no spanning-tree mode mst**.
3. **exit**
4. (Optional) **show running-config spanning-tree all**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**spanning-tree mode mst** or **no spanning-tree mode mst**. **Example:** `switch(config)# spanning-tree mode mst`	• **spanning-tree mode mst** Enables MST on the device. • **no spanning-tree mode mst** Disables MST on the device and returns you to Rapid PVST+.

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

Step 4	(Optional) **show running-config spanning-tree all** **Example:** `switch# show running-config spanning-tree all`	Displays the currently running STP configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to enable MST on the device:

```
switch# config t
switch(config)# spanning-tree mode mst
switch(config)# exit
switch#
```

Entering MST Configuration Mode

You enter MST configuration mode to configure the MST name, VLAN-to-instance mapping, and MST revision number on the device.

If two or more devices are in the same MST region, they must have the identical MST name, VLAN-to-instance mapping, and MST revision number.

Note Each command reference line creates its pending regional configuration in MST configuration mode. In addition, the pending region configuration starts with the current region configuration.

SUMMARY STEPS

1. **config t**
2. **spanning-tree mst configuration** or **no spanning-tree mst configuration**
3. **exit** or **abort**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**spanning-tree mst configuration** or **no spanning-tree mst configuration**
	Example:
	`switch(config)# spanning-tree mst configuration`
	`switch(config-mst)#`
		• **spanning-tree mst configuration**
		Enters MST configuration submode on the system. You must be in the MST configuration submode to assign the MST configuration parameters, as follows:
		• MST name
		• VLAN-to-MST instance mapping
		• MST revision number
		• **no spanning-tree mst configuration**
		Returns the MST region configuration to the following default values:
		• The region name is an empty string.
		• No VLANs are mapped to any MST instance (all VLANs are mapped to the CIST instance).

	Command or Action	Purpose
		• The revision number is 0.

Step 3	**exit** or **abort**
	Example:
	`switch(config-mst)# exit`
	`switch(config)#`
		• **exit**
		Commits all the changes and exits MST configuration submode.
		• **abort**
		Exits the MST configuration submode without committing any of the changes.

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

Example

This example shows how to enter the MST configuration submode on the device:

```
switch# config t
switch(config)# spanning-tree mst configuration
switch(config-mst)# exit
switch(config)#
```

Specifying the MST Name

You can configure a region name on the bridge. If two or more bridges are in the same MST region, they must have the identical MST name, VLAN-to-instance mapping, and MST revision number.

SUMMARY STEPS

1. **config t**
2. **spanning-tree mst configuration**
3. **name** *name*
4. **exit** or **abort**
5. (Optional) **show spanning-tree mst configuration**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**config t** **Example:**	Enters configuration mode.

	Command or Action	Purpose
	`switch# config t` `switch(config)#`	

Step 2	**spanning-tree mst configuration**
	Example:
	`switch(config)# spanning-tree mst configuration`
	`switch(config-mst)#`
		Enters MST configuration submode.

Step 3	**name** *name* **Example:** `switch(config-mst)# name accounting`	Specifies the name for the MST region. The *name* string has a maximum length of 32 characters and is case sensitive. The default is an empty string.

Step 4	**exit** or **abort**
	Example:
	`switch(config-mst)# exit`
	`switch(config)#`
		• **exit**
		Commits all the changes and exits MST configuration submode.
		• **abort**
		Exits the MST configuration submode without committing any of the changes.

Step 5	(Optional) **show spanning-tree mst configuration** **Example:** `switch# show spanning-tree mst configuration`	Displays the MST configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to set the name of the MST region:

```
switch# config t
switch(config)#  spanning-tree mst configuration
switch(config-mst)# name accounting
switch(config-mst)# exit
switch(config)#
```

Specifying the MST Configuration Revision Number

You configure the revision number on the bridge. If two or more bridges are in the same MST region, they must have the identical MST name, VLAN-to-instance mapping, and MST revision number.

SUMMARY STEPS

1. **config t**
2. **spanning-tree mst configuration**

3. **revision** *version*
4. **exit** or **abort**
5. (Optional) **show spanning-tree mst configuration**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**spanning-tree mst configuration**
	Example:
	`switch(config)# spanning-tree mst configuration`
	`switch(config-mst)#`
		Enters MST configuration submode.

Step 3	**revision** *version* **Example:** `switch(config-mst)# revision 5`	Specifies the revision number for the MST region. The range is from 0 to 65535, and the default value is 0.

Step 4	**exit** or **abort**
	Example:
	`switch(config-mst)# exit`
	`switch(config)#`
		• **exit**
		Commits all the changes and exits MST configuration submode.
		• **abort**
		Exits the MST configuration submode without committing any of the changes.

Step 5	(Optional) **show spanning-tree mst configuration** **Example:** `switch# show spanning-tree mst configuration`	Displays the MST configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the revision number of the MSTI region to 5:

```
switch# config t
switch(config)# spanning-tree mst configuration
```

```
switch(config-mst)# revision 5
switch(config-mst)#
```

Specifying the Configuration on an MST Region

If two or more devices are to be in the same MST region, they must have the same VLAN-to-instance mapping, the same configuration revision number, and the same MST name.

A region can have one member or multiple members with the same MST configuration; each member must be capable of processing IEEE 802.1w RSTP BPDUs. There is no limit to the number of MST regions in a network, but each region can support only up to 65 MST instances. You can assign a VLAN to only one MST instance at a time.

SUMMARY STEPS

1. **config t**
2. **spanning-tree mst configuration**
3. **instance** *instance-id* **vlan** *vlan-range*
4. **name** *name*
5. **revision** *version*
6. **exit** or **abort**
7. **show spanning-tree mst configuration**
8. **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**spanning-tree mst configuration**
	Example:
	`switch(config)# spanning-tree mst configuration`
	`switch(config-mst)#`
		Enters MST configuration submode.

Step 3	**instance** *instance-id* **vlan** *vlan-range* **Example:** `switch(config-mst)# instance 1 vlan 10-20`	Maps VLANs to an MST instance as follows: • For *instance-id*, the range is from 1 to 4094. • For **vlan** *vlan-range*, the range is from 1 to 3967. When you map VLANs to an MST instance, the mapping is incremental, and the VLANs specified in the command are added to or removed from the VLANs that were previously mapped.

	Command or Action	Purpose
		To specify a VLAN range, enter a hyphen; for example, enter the **instance 1 vlan 1-63** command to map VLANs 1 through 63 to MST instance 1. To specify a VLAN series, enter a comma; for example, enter the **instance 1 vlan 10, 20, 30** command to map VLANs 10, 20, and 30 to MST instance 1.
Step 4	**name** *name* **Example:** `switch(config-mst)# name region1`	Specifies the instance name. The name string has a maximum length of 32 characters and is case sensitive.
Step 5	**revision** *version* **Example:** `switch(config-mst)# revision 1`	Specifies the configuration revision number. The range is from 0 to 65535.

Step 6	**exit** or **abort**
	Example:
	`switch(config-mst)# exit`
	`switch(config)#`
		• **exit**
		Commits all the changes and exits MST configuration submode.
		• **abort**
		Exits the MST configuration submode without committing any of the changes.

Step 7	**show spanning-tree mst configuration** **Example:** `switch# show spanning-tree mst configuration`	(Optional) Displays the MST configuration.
Step 8	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to enter MST configuration mode, map VLANs 10 to 20 to MST instance 1, name the region *region1*, set the configuration revision to 1, display the pending configuration, apply the changes, and return to global configuration mode:

```
switch# config t
switch(config)# spanning-tree mst configuration
switch(config-mst)# instance 1 vlan 10-20
switch(config-mst)# name region1
switch(config-mst)# revision 1
switch(config-mst#) exit
switch(config)# show spanning-tree mst configuration

Name      [region1]
Revision  1
Instances configured 2
```

```
Instance  Vlans Mapped
--------  ---------------------
0         1-9,21-4094
1         10-20
-------------------------------
switch(config)#
```

Mapping or Unmapping a VLAN to an MST Instance - CLI Version

If two or more bridges are to be in the same MST region, they must have the identical MST name, VLAN-to-instance mapping, and MST revision number.

You cannot map VLANs 3968 to 4095 to an MST instance. These VLANs are reserved for internal use by the device.

Note When you change the VLAN-to-MSTI mapping, the system reconverges MST.

Note You cannot disable an MSTI.

SUMMARY STEPS

1. **config t**
2. **spanning-tree mst configuration**
3. **instance** *instance-id* **vlan** *vlan-range* or **no instance** *instance-id* **vlan** *vlan-range*
4. **exit** or **abort**
5. (Optional) **show spanning-tree mst configuration**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**spanning-tree mst configuration**
	Example:
	`switch(config)# spanning-tree mst configuration`
	`switch(config-mst)#`
		Enters MST configuration submode.

	Command or Action	Purpose
Step 3	**instance** *instance-id* **vlan** *vlan-range* or **no instance** *instance-id* **vlan** *vlan-range* **Example:** `switch(config-mst)# instance 3 vlan 200`	• **instance** *instance-id* **vlan** *vlan-range* Maps VLANs to an MST instance as follows: • For *instance_id*, the range is from 1 to 4094. Instance 0 is reserved for the IST for each MST region. • For *vlan-range*, the range is from 1 to 3967. When you map VLANs to an MSTI, the mapping is incremental, and the VLANs specified in the command are added to or removed from the VLANs that were previously mapped. • **no instance** *instance-id* **vlan** *vlan-range* Deletes the specified instance and returns the VLANs to the default MSTI, which is the CIST.

Step 4	**exit** or **abort**
	Example:
	`switch(config-mst)# exit`
	`switch(config)#`
		• **exit**
		Commits all the changes and exits MST configuration submode.
		• **abort**
		Exits the MST configuration submode without committing any of the changes.

Step 5	(Optional) **show spanning-tree mst configuration** **Example:** `switch# show spanning-tree mst configuration`	Displays the MST configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to map VLAN 200 to MSTI 3:

```
switch# config t
switch(config)# spanning-tree mst configuration
switch(config-mst)# instance 3 vlan 200
switch(config-mst)# exit
switch(config)#
```

Configuring the Root Bridge

You can configure the device to become the MST root bridge.

The **spanning-tree vlan** *vlan_ID* **primary root** command fails if the value required to be the root bridge is less than 4096. If the software cannot lower the bridge priority any lower, the device returns the following message:

```
Error: Failed to set root bridge for VLAN 1
It may be possible to make the bridge root by setting the priority
for some (or all) of these instances to zero.
```

Note The root bridge for each MSTI should be a backbone or distribution device. Do not configure an access device as the spanning tree primary root bridge.

Enter the **diameter** keyword, which is available only for MSTI 0 (or the IST), to specify the Layer 2 network diameter (that is, the maximum number of Layer 2 hops between any two end stations in the Layer 2 network). When you specify the network diameter, the device automatically sets an optimal hello time, forward-delay time, and maximum-age time for a network of that diameter, which can significantly reduce the convergence time. You can enter the **hello** keyword to override the automatically calculated hello time.

Note With the device configured as the root bridge, do not manually configure the hello time, forward-delay time, and maximum-age time using the **spanning-tree mst hello-timespanning-tree mst forward-time**, and **spanning-tree mst max-age** global configuration commands.

SUMMARY STEPS

1. **config t**
2. **spanning-tree mst** *instance-id* **root** {**primary** | **secondary**} [**diameter** *dia* [**hello-time** *hello-time*]] or **no spanning-tree mst** *instance-id* **root**
3. **exit** or **abort**
4. (Optional) **show spanning-tree mst**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**spanning-tree mst** *instance-id* **root** {**primary** | **secondary**} [**diameter** *dia* [**hello-time** *hello-time*]] or **no spanning-tree mst** *instance-id* **root** **Example:** `switch(config)# spanning-tree mst 5 root primary`	• **spanning-tree mst** *instance-id* **root** {**primary** | **secondary**} [**diameter** *dia* [**hello-time** *hello-time*]] Configures a device as the root bridge as follows: • For *instance-id*, specify a single instance, a range of instances separated by a hyphen, or a series of

	Command or Action	Purpose
		instances separated by a comma. The range is from 1 to 4094. • For **diameter** *net-diameter*, specify the maximum number of Layer 2 hops between any two end stations. The default is 7. This keyword is available only for MST instance 0. • For **hello-time** *seconds*, specify the interval in seconds between the generation of configuration messages by the root bridge. The range is from 1 to 10 seconds; the default is 2 seconds. • **no spanning-tree mst** *instance-id* **root** Returns the switch priority, diameter, and hello time to default values.

Step 3	**exit** or **abort**
	Example:
	`switch(config)# exit`
	`switch#`
		• **exit**
		Commits all the changes and exits MST configuration submode.
		• **abort**
		Exits the MST configuration submode without committing any of the changes.

Step 4	(Optional) **show spanning-tree mst** **Example:** `switch# show spanning-tree mst`	Displays the MST configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the device as the root switch for MSTI 5:

```
switch# config t
switch(config)# spanning-tree mst 5 root primary
switch(config)# exit
switch(config)#
```

Configuring an MST Secondary Root Bridge

You use this command on more than one device to configure multiple backup root bridges. Enter the same network diameter and hello-time values that you used when you configured the primary root bridge with the **spanning-tree mst root primary** global configuration command.

SUMMARY STEPS

1. **config t**
2. **spanning-tree mst** *instance-id* **root** {**primary** | **secondary**} [**diameter** *dia*[**hello-time** *hello-time*]] or **no spanning-tree mst** *instance-id* **root**
3. **exit**
4. (Optional) **show spanning-tree mst**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**spanning-tree mst** *instance-id* **root** {**primary** | **secondary**} [**diameter** *dia*[**hello-time** *hello-time*]] or **no spanning-tree mst** *instance-id* **root** **Example:** `switch(config)# spanning-tree mst 5 root secondary`	• **spanning-tree mst** *instance-id* **root** {**primary** | **secondary**} [**diameter** *dia*[**hello-time** *hello-time*]] Configures a device as the secondary root bridge as follows: • For *instance-id*, you can specify a single instance, a range of instances separated by a hyphen, or a series of instances separated by a comma. The range is from 1 to 4094. • For **diameter** *net-diameter*, specify the maximum number of Layer 2 hops between any two end stations. The default is 7. This keyword is available only for MST instance 0. • For **hello-time** *seconds*, specify the interval in seconds between the generation of configuration messages by the root bridge. The range is from 1 to 10 seconds; the default is 2 seconds. • **no spanning-tree mst** *instance-id* **root** Returns the switch priority, diameter, and hello-time to default values.

	Command or Action	Purpose

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

Step 4	(Optional) **show spanning-tree mst** **Example:** `switch# show spanning-tree mst`	Displays the MST configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the device as the secondary root switch for MSTI 5:

```
switch# config t
switch(config)# spanning-tree mst 5 root secondary
switch(config)# exit
switch#
```

Configuring the MST Switch Priority

You can configure the switch priority for an MST instance so that it is more likely that the specified device is chosen as the root bridge.

Note Be careful when using the **spanning-tree mst priority** command. For most situations, we recommend that you enter the **spanning-tree mst root primary** and the **spanning-tree mst root secondary** global configuration commands to modify the switch priority.

SUMMARY STEPS

1. **config t**
2. **spanning-tree mst** *instance-id* **priority** *priority-value*
3. **exit**
4. (Optional) **show spanning-tree mst**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**spanning-tree mst** *instance-id* **priority** *priority-value* **Example:** `switch(config)# spanning-tree mst 5 priority 4096`	Configures a device priority as follows: • For *instance-id*, you can specify a single instance, a range of instances separated by a hyphen, or a series of instances separated by a comma. The range is from 1 to 4094. • For *priority-value* the range is from 0 to 61440 in increments of 4096; the default is 32768. A lower number indicates that the device will most likely be chosen as the root bridge. Priority values are 0, 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768, 36864, 40960, 45056, 49152, 53248, 57344, and 61440. The system rejects all other values.

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

Step 4	(Optional) **show spanning-tree mst** **Example:** `switch# show spanning-tree mst`	Displays the MST configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the priority of the bridge to 4096 for MSTI 5:

```
switch# config t
switch(config)# spanning-tree mst 5 priority 4096
switch(config)# exit
switch#
```

Configuring the MST Port Priority

If a loop occurs, MST uses the port priority when selecting an interface to put into the forwarding state. You can assign lower priority values to interfaces that you want selected first and higher priority values to the interface that you want selected last. If all interfaces have the same priority value, MST puts the interface with the lowest interface number in the forwarding state and blocks the other interfaces.

SUMMARY STEPS

1. **config t**
2. **interface** {{*type slot/port*} | {**port-channel** *number*}}
3. **spanning-tree mst** *instance-id* **port-priority** *priority*
4. **exit**
5. (Optional) **show spanning-tree mst**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**interface** {{*type slot/port*} | {**port-channel** *number*}}
	Example:
	`switch(config)# interface ethernet 3/1`
	`switch(config-if)#`
		Specifies an interface to configure, and enters interface configuration mode.

Step 3	**spanning-tree mst** *instance-id* **port-priority** *priority* **Example:** `switch(config-if)# spanning-tree mst 3 port-priority 64`	Configures the port priority as follows: • For *instance-id*, you can specify a single MSTI, a range of MSTIs separated by a hyphen, or a series of MSTIs separated by a comma. The range is from 1 to 4094. • For *priority*, the range is from 0 to 224 in increments of 32. The default is 128. A lower number indicates a higher priority. The priority values are 0, 32, 64, 96, 128, 160, 192, and 224. The system rejects all other values.

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

	Command or Action	Purpose
Step 5	(Optional) **show spanning-tree mst** **Example:** `switch# show spanning-tree mst`	Displays the MST configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to set the MST interface port priority for MSTI 3 on Ethernet port 3/1 to 64:

```
switch# config t
switch(config)# interface ethernet 3/1
switch(config-if)# spanning-tree mst 3 port-priority 64
switch(config-if)# exit
switch(config)#
```

Configuring the MST Port Cost

The MST port cost default value is derived from the media speed of an interface. If a loop occurs, MST uses the cost when selecting an interface to put in the forwarding state. You can assign lower cost values to interfaces that you want selected first and higher cost to interfaces values that you want selected last. If all interfaces have the same cost value, MST puts the interface with the lowest interface number in the forwarding state and blocks the other interfaces.

Note MST uses the long path-cost calculation method.

SUMMARY STEPS

1. **config t**
2. **interface** {{*type slot/port*} | {**port-channel** *number*}}
3. **spanning-tree mst** *instance-id* **cost** {*cost* | *auto*}
4. **exit**
5. (Optional) **show spanning-tree mst**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**interface** {{*type slot/port*} | {**port-channel** *number*}}
	Example:
	`switch# config t`
	`switch(config)# interface ethernet 3/1`
	`switch(config-if)#`
		Specifies an interface to configure, and enters interface configuration mode.

Step 3	**spanning-tree mst** *instance-id* **cost** {*cost* | *auto*} **Example:** `switch(config-if)# spanning-tree mst 4 cost 17031970`	Configures the cost. If a loop occurs, MST uses the path cost when selecting an interface to place into the forwarding state. A lower path cost represents higher-speed transmission as follows: • For *instance-id*, you can specify a single instance, a range of instances separated by a hyphen, or a series of instances separated by a comma. The range is from 1 to 4094. • For *cost*, the range is from 1 to 200000000. The default value is **auto**, which is derived from the media speed of the interface.

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

Step 5	(Optional) **show spanning-tree mst** **Example:** `switch# show spanning-tree mst`	Displays the MST configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to set the MST interface port cost on Ethernet 3/1 for MSTI 4:

```
switch# config t
switch(config)# interface ethernet 3/1
switch(config-if)# spanning-tree mst 4 cost 17031970
switch(config-if)# exit
switch(config)#
```

Configuring the MST Hello Time

You can configure the interval between the generation of configuration messages by the root bridge for all instances on the device by changing the hello time.

Note Be careful when using the **spanning-tree mst hello-time** command. For most situations, we recommend that you enter the **spanning-tree mst** *instance-id* **root primary** and the **spanning-tree mst** *instance-id* **root secondary** global configuration commands to modify the hello time.

SUMMARY STEPS

1. **config t**
2. **spanning-tree mst hello-time** *seconds*
3. **exit**
4. (Optional) **show spanning-tree mst**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**spanning-tree mst hello-time** *seconds* **Example:** `switch(config)# spanning-tree mst hello-time 1`	Configures the hello time for all MST instances. The hello time is the interval between the generation of configuration messages by the root bridge. These messages mean that the device is alive. For *seconds*, the range is from 1 to 10, and the default is 2 seconds.

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

Step 4	(Optional) **show spanning-tree mst** **Example:**	Displays the MST configuration.

	Command or Action	Purpose
	`switch# show spanning-tree mst`	
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the hello time of the device to 1 second:

```
switch# config t
switch(config)# spanning-tree mst hello-time 1
switch(config)# exit
switch#
```

Configuring the MST Forwarding-Delay Time

You can set the forward delay timer for all MST instances on the device with one command.

SUMMARY STEPS

1. **config t**
2. **spanning-tree mst forward-time** *seconds*
3. **exit**
4. (Optional) **show spanning-tree mst**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**spanning-tree mst forward-time** *seconds* **Example:** `switch(config)# spanning-tree mst forward-time 10`	Configures the forward time for all MST instances. The forward delay is the number of seconds that a port waits before changing from its spanning tree blocking and learning states to the forwarding state. For *seconds*, the range is from 4 to 30, and the default is 15 seconds.
Step 3	**exit** **Example:**	Exits configuration mode.

	Command or Action	Purpose
	`switch(config)# exit` `switch#`	
Step 4	(Optional) **show spanning-tree mst** **Example:** `switch# show spanning-tree mst`	Displays the MST configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the forward-delay time of the device to 10 seconds:

```
switch# config t
switch(config)# spanning-time mst forward-time 10
switch(config)# exit
switch#
```

Configuring the MST Maximum-Aging Time

You can set the maximum-aging timer for all MST instances on the device with one command (the maximum age time only applies to the IST).

The maximum-aging timer is the number of seconds that a device waits without receiving spanning tree configuration messages before attempting a reconfiguration.

SUMMARY STEPS

1. **config t**
2. **spanning-tree mst max-age** *seconds*
3. **exit**
4. (Optional) **show spanning-tree mst**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

	Command or Action	Purpose
Step 2	**spanning-tree mst max-age** *seconds* **Example:** `switch(config)# spanning-tree mst max-age 40`	Configures the maximum-aging time for all MST instances. The maximum-aging time is the number of seconds that a device waits without receiving spanning tree configuration messages before attempting a reconfiguration. For *seconds*, the range is from 6 to 40, and the default is 20 seconds.

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

Step 4	(Optional) **show spanning-tree mst** **Example:** `switch# show spanning-tree mst`	Displays the MST configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the maximum-aging timer of the device to 40 seconds:

```
switch# config t
switch(config)# spanning-tree mst max-age 40
switch(config)# exit
switch#
```

Configuring the MST Maximum-Hop Count

You can configure the maximum hops inside the region and apply it to the IST and all MST instances in that region. MST uses the path cost to the IST regional root and a hop-count mechanism similar to the IP time-to-live (TTL) mechanism. The hop count achieves the same result as the message-age information (triggers a reconfiguration).

SUMMARY STEPS

1. **config t**
2. **spanning-tree mst max-hops** *hop-count*
3. **exit**
4. (Optional) **show spanning-tree mst**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**spanning-tree mst max-hops** *hop-count* **Example:** `switch(config)# spanning-tree mst max-hops 40`	Specifies the number of hops in a region before the BPDU is discarded and the information held for a port is aged. For *hop-count*, the range is from 1 to 255, and the default value is 20 hops.

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

Step 4	(Optional) **show spanning-tree mst** **Example:** `switch# show spanning-tree mst`	Displays the MST configuration.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to set the maximum hops to 40:

```
switch# config t
switch(config)# spanning-tree mst max-hops 40
switch(config)# exit
switch#
```

Configuring an Interface to Proactively Send Prestandard MSTP Messages - CLI Version

By default, interfaces on a device running MST send prestandard, rather than standard, MSTP messages after they receive a prestandard MSTP message from another interface. You can configure the interface to proactively send prestandard MSTP messages. That is, the specified interface would not have to wait to receive a prestandard MSTP message; the interface with this configuration always sends prestandard MSTP messages.

SUMMARY STEPS

1. **config t**
2. **interface** *type slot/port*
3. **spanning-tree mst pre-standard**
4. **exit**
5. (Optional) **show spanning-tree mst**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)#`
		Specifies the interface to configure and enters the interface configuration mode.

Step 3	**spanning-tree mst pre-standard** **Example:** `switch(config-if)# spanning-tree mst pre-standard`	Specifies that the interface always sends MSTP messages in the prestandard format, rather than in the MSTP standard format.

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

Step 5	(Optional) **show spanning-tree mst** **Example:** `switch# show spanning-tree mst`	Displays the MST configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to set the MST interface so that it always sends MSTP messages in the prestandard format:

```
switch# config t
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree mst pre-standard
switch(config-if)# exit
switch(config)#
```

Specifying the Link Type for MST - CLI Version

Rapid connectivity (802.1w standard) is established only on point-to-point links. By default, the link type is controlled from the duplex mode of the interface. A full-duplex port is considered to have a point-to-point connection; a half-duplex port is considered to have a shared connection.

If you have a half-duplex link physically connected point to point to a single port on a remote device, you can override the default setting on the link type and enable rapid transitions.

If you set the link to shared, STP falls back to 802.1D.

SUMMARY STEPS

1. **config t**
2. **interface** *type slot/port*
3. **spanning-tree link-type** {*auto* | *point-to-point* | *shared*}
4. **exit**
5. (Optional) **show spanning-tree**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)#`
		Specifies the interface to configure and enters the interface configuration mode.

Step 3	**spanning-tree link-type** {*auto* | *point-to-point* | *shared*} **Example:** `switch(config-if)# spanning-tree link-type point-to-point`	Configures the link type to be either a point-to-point link or shared link. The system reads the default value from the device connection, as follows: half duplex links are shared and full-duplex links are point to point. If the link type is shared, the STP falls back to 802.1D. The default is auto, which sets the link type based on the duplex setting of the interface.

	Command or Action	Purpose

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

Step 5	(Optional) **show spanning-tree** **Example:** `switch# show spanning-tree`	Displays the STP configuration.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the link type as a point-to-point link:

```
switch# config t
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree link-type point-to-point
switch(config-if)# exit
switch(config)#
```

Reinitializing the Protocol for MST

An MST bridge can detect that a port is at the boundary of a region when it receives a legacy BPDU or an MST BPDU that is associated with a different region. However, the STP protocol migration cannot determine whether the legacy device, which is a device that runs only IEEE 802.1D, has been removed from the link unless the legacy device is the designated switch. Enter this command to reinitialize the protocol negotiation (force the renegotiation with neighboring devices) on the entire device or on specified interfaces.

SUMMARY STEPS

1. **clear spanning-tree detected-protocol** [**interface** *interface* [*interface-num* | *port-channel*]]

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**clear spanning-tree detected-protocol** [**interface** *interface* [*interface-num*	*port-channel*]] **Example:** `switch# clear spanning-tree detected-protocol`	Reintializes MST on an entire device or specified interfaces.

Example

This example shows how to reinitialize MST on the Ethernet interface on slot 2, port 8:

```
switch# clear spanning-tree detected-protocol interface ethernet 2/8
```

Verifying the MST Configuration

To display MST configuration information, perform one of the following tasks:

Command	Purpose
show running-config spanning-tree [**all**]	Displays STP information.
show spanning-tree mst configuration	Displays MST information.
show spanning-tree mst [**detail**]	Displays information about MST instances.
show spanning-tree mst *instance-id* [**detail**]	Displays information about the specified MST instance.
show spanning-tree mst *instance-id* **interface** {**ethernet** *slot/port* | **port-channel** *channel-number*} [**detail**]	Displays MST information for the specified interface and instance.
show spanning-tree summary	Displays summary STP information.
show spanning-tree detail	Displays detailed STP information.
show spanning-tree {**vlan** *vlan-id* | **interface** {[**ethernet** *slot/port*] | [**port-channel** *channel-number*]}} [**detail**]	Displays STP information per VLAN and interface.
show spanning-tree vlan *vlan-id* **bridge**	Displays information on the STP bridge.

Displaying and Clearing MST Statistics -- CLI Version

To display MST configuration information, perform one of the following tasks:

Command	Purpose
clear spanning-tree counters [**interface** *type slot/port* | **vlan***vlan-id*]	Clears the counters for STP.
show spanning-tree {**vlan** *vlan-id* | **interface** {[**ethernet** *slot/port*] | [**port-channel***channel-number*]}} **detail**	Displays information about STP by interface or VLAN including BPDUs sent and received.

MST Example Configuration

The following example shows how to configure MST:

```
switch# configure terminal
switch(config)# spanning-tree mode mst
switch(config)# spanning-tree port type edge bpduguard default
switch(config)# spanning-tree port type edge bpdufilter default
switch(config)# spanning-tree port type network default
switch(config)# spanning-tree mst 0-64 priority 24576
switch(config)# spanning-tree mst configuration
switch(config-mst)# name cisco_region_1
switch(config-mst)# revision 2
switch(config-mst)# instance 1 vlan 1-21
switch(config-mst)# instance 2 vlan 22-42
switch(config-mst)# instance 3 vlan 43-63
switch(config-mst)# instance 4 vlan 64-84
switch(config-mst)# instance 5 vlan 85-105
switch(config-mst)# instance 6 vlan 106-126
switch(config-mst)# instance 6 vlan 106-126
switch(config-mst)# instance 7 vlan 127-147
switch(config-mst)# instance 8 vlan 148-168
switch(config-mst)# instance 9 vlan 169-189
switch(config-mst)# instance 10 vlan 190-210
switch(config-mst)# instance 11 vlan 211-231
switch(config-mst)# instance 12 vlan 232-252
switch(config-mst)# instance 13 vlan 253-273
switch(config-mst)# instance 14 vlan 274-294
switch(config-mst)# instance 15 vlan 295-315
switch(config-mst)# instance 16 vlan 316-336
switch(config-mst)# instance 17 vlan 337-357
switch(config-mst)# instance 18 vlan 358-378
switch(config-mst)# instance 19 vlan 379-399
switch(config-mst)# instance 20 vlan 400-420
switch(config-mst)# instance 21 vlan 421-441
switch(config-mst)# instance 22 vlan 442-462
switch(config-mst)# instance 23 vlan 463-483
switch(config-mst)# instance 24 vlan 484-504
switch(config-mst)# instance 25 vlan 505-525
switch(config-mst)# instance 26 vlan 526-546
switch(config-mst)# instance 27 vlan 547-567
switch(config-mst)# instance 28 vlan 568-588
switch(config-mst)# instance 29 vlan 589-609
switch(config-mst)# instance 30 vlan 610-630
switch(config-mst)# instance 31 vlan 631-651
switch(config-mst)# instance 32 vlan 652-672
switch(config-mst)# instance 33 vlan 673-693
switch(config-mst)# instance 34 vlan 694-714
switch(config-mst)# instance 35 vlan 715-735
switch(config-mst)# instance 36 vlan 736-756
switch(config-mst)# instance 37 vlan 757-777
switch(config-mst)# instance 38 vlan 778-798
switch(config-mst)# instance 39 vlan 799-819
switch(config-mst)# instance 40 vlan 820-840
switch(config-mst)# instance 41 vlan 841-861
switch(config-mst)# instance 42 vlan 862-882
switch(config-mst)# instance 43 vlan 883-903
switch(config-mst)# instance 44 vlan 904-924
switch(config-mst)# instance 45 vlan 925-945
switch(config-mst)# instance 46 vlan 946-966
```

```
switch(config-mst)# instance 47 vlan 967-987
switch(config-mst)# instance 48 vlan 988-1008
switch(config-mst)# instance 49 vlan 1009-1029
switch(config-mst)# instance 50 vlan 1030-1050
switch(config-mst)# instance 51 vlan 1051-1071
switch(config-mst)# instance 52 vlan 1072-1092
switch(config-mst)# instance 53 vlan 1093-1113
switch(config-mst)# instance 54 vlan 1114-1134
switch(config-mst)# instance 55 vlan 1135-1155
switch(config-mst)# instance 56 vlan 1156-1176
switch(config-mst)# instance 57 vlan 1177-1197
switch(config-mst)# instance 58 vlan 1198-1218
switch(config-mst)# instance 59 vlan 1219-1239
switch(config-mst)# instance 60 vlan 1240-1260
switch(config-mst)# instance 61 vlan 1261-1281
switch(config-mst)# instance 62 vlan 1282-1302
switch(config-mst)# instance 63 vlan 1303-1323
switch(config-mst)# instance 64 vlan 1324-1344
switch(config-mst)# exit

switch(config)# interface ethernet 3/1
switch(config-if)# switchport
switch(config-if)# no shutdown
switch(config-if)# spanning-tree port type edge
switch(config-if)# exit

switch(config)# interface ethernet 3/2
switch(config-if)# switchport
switch(config-if)# switchport mode trunk
switch(config-if)# no shutdown
switch(config-if)# spanning-tree guard root
switch(config-if)# exit
switch(config)#
```

Additional References for MST -- CLI Version

Related Documents

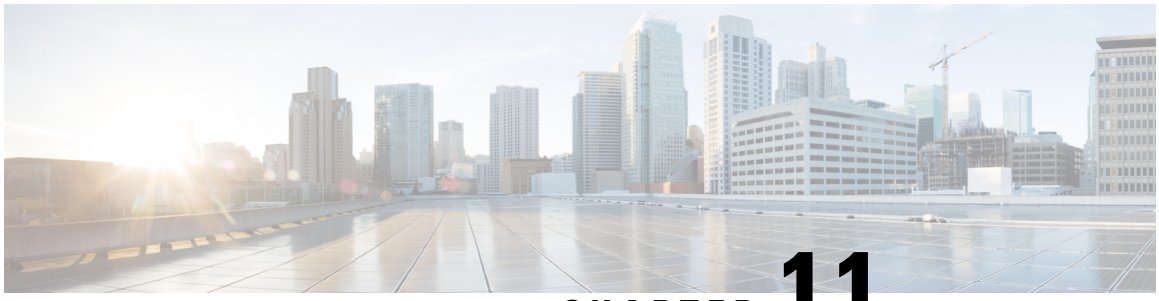
Related Topic	Document Title
Layer 2 interfaces	*Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*
NX-OS fundamentals	*Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide*
High availability	*Cisco Nexus 9000 Series High Availability and Redundancy Guide*
System management	*Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*

Standards

Standards	Title
IEEE 802.1Q-2006 (formerly known as IEEE 802.1s), IEEE 802.1D-2004 (formerly known as IEEE 802.1w), IEEE 802.1D, IEEE 802.1t	—

MIBS

MIBs	MIBs Link
CISCO-STP-EXTENSION-MIB BRIDGE-MIB	To locate and download MIBs, go to the following URL: https://cisco.github.io/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html



CHAPTER 11

Configuring STP Extensions Using Cisco NX-OS

- Information About STP Extensions, on page 175
- Prerequisites for STP Extensions, on page 181
- Guidelines and Limitations for Configuring STP Extensions, on page 181
- Default Settings for STP Extensions, on page 183
- Configuring STP Extensions Steps, on page 183
- Verifying the STP Extension Configuration, on page 202
- Configuration Examples for STP Extension, on page 203
- Additional References for STP Extensions -- CLI Version, on page 203

Information About STP Extensions

Note See the *Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*, for information on creating Layer 2 interfaces.

Cisco has added extensions to STP that enhances loop prevention, protects against some possible user misconfigurations, and provides better control over the protocol parameters. Although, in some cases, similar functionality may be incorporated into the IEEE 802.1w Rapid Spanning Tree Protocol (RSTP) standard, we recommend using these extensions. All of these extensions, except PVST Simulation, can be used with both Rapid PVST+ and MST. You use PVST Simulation only with MST.

The available extensions are spanning tree edge ports (which supply the functionality previously known as PortFast), Bridge Assurance, BPDU Guard, BPDU Filtering, Loop Guard, Root Guard, and PVT Simulation. Many of these features can be applied either globally or on specified interfaces.

Note Spanning tree is used to refer to IEEE 802.1w and IEEE 802.1s. If the text is discussing the IEEE 802.1D Spanning Tree Protocol, 802.1D is stated specifically.

STP Port Types

You can configure a spanning tree port as an edge port, a network port, or a normal port. A port can be in only one of these states at a given time. The default spanning tree port type is normal.

Edge ports, which are connected to Layer 2 hosts, can be either an access port or a trunk port.

Note If you configure a port connected to a Layer 2 switch or bridge as an edge port, you might create a bridging loop.

Network ports are connected only to Layer 2 switches or bridges.

Note If you mistakenly configure ports that are connected to Layer 2 hosts, or edge devices, as spanning tree network ports, those ports will automatically move into the blocking state.

STP Edge Ports

You connect STP edge ports only to Layer 2 hosts. The edge port interface immediately transitions to the forwarding state, without moving through the blocking or learning states. (This immediate transition was previously configured as the Cisco-proprietary feature PortFast.)

Interfaces that are connected to Layer 2 hosts should not receive STP bridge protocol data units (BPDUs).

Bridge Assurance

You can use Bridge Assurance to protect against certain problems that can cause bridging loops in the network. Specifically, you use Bridge Assurance to protect against a unidirectional link failure or other software failure and a device that continues to forward data traffic when it is no longer running the spanning tree algorithm.

Note Bridge Assurance is supported only by Rapid PVST+ and MST.

Bridge Assurance takes 2 seconds to kick in on a regular link and ~84 seconds on a VPC peer-link.

Bridge Assurance is enabled by default and can only be disabled globally. Also, Bridge Assurance can be enabled only on spanning tree network ports that are point-to-point links. Finally, both ends of the link must have Bridge Assurance enabled. If the device on one side of the link has Bridge Assurance enabled and the device on the other side either does not support Bridge Assurance or does not have this feature enabled, the connecting port is blocked.

With Bridge Assurance enabled, BPDUs are sent out on all operational network ports, including alternate and backup ports, for each hello time period. If the port does not receive a BPDU for a specified period, the port moves into the blocking state and is not used in the root port calculation. Once that port receives a BPDU, it resumes the normal spanning tree transitions.

Figure 15: Network with Normal STP Topology

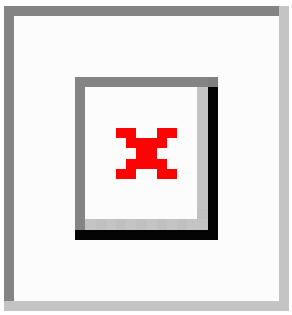
This figure shows a normal STP topology. 

Figure 16: Network Problem without Running Bridge Assurance

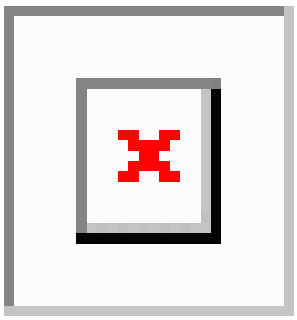
This figure demonstrates a potential network problem when the device fails and you are not running Bridge Assurance. 

Figure 17: Network STP Topology Running Bridge Assurance

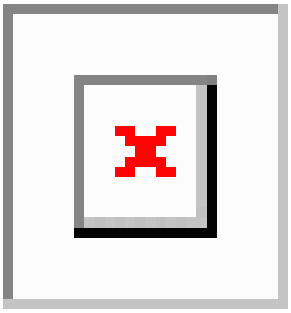
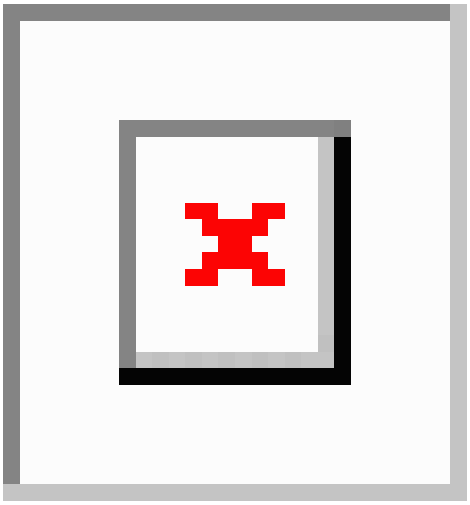
This figure shows the network with Bridge Assurance enabled, and the STP topology progressing normally with bidirectional BPDUs issuing from every STP network port. 

Figure 18: Network Problem Averted with Bridge Assurance Enabled

This figure shows how the potential network problem does not happen when you have Bridge Assurance enabled on your network.



BPDU Guard

Enabling BPDU Guard shuts down that interface if a BPDU is received.

You can configure BPDU Guard at the interface level. When configured at the interface level, BPDU Guard shuts the port down as soon as the port receives a BPDU, regardless of the port type configuration.

When you configure BPDU Guard globally, it is effective only on operational spanning tree edge ports. In a valid configuration, Layer 2 LAN edge interfaces do not receive BPDUs. A BPDU that is received by an edge Layer 2 LAN interface signals an invalid configuration, such as the connection of an unauthorized device. BPDU Guard, when enabled globally, shuts down all spanning tree edge ports when they receive a BPDU.

BPDU Guard provides a secure response to invalid configurations, because you must manually put the Layer 2 LAN interface back in service after an invalid configuration.

Note When enabled globally, BPDU Guard applies to all operational spanning tree edge interfaces.

BPDU Filtering

You can use BPDU Filtering to prevent the device from sending or even receiving BPDUs on specified ports.

When configured globally, BPDU Filtering applies to all operational spanning tree edge ports. You should connect edge ports only to hosts, which typically drop BPDUs. If an operational spanning tree edge port receives a BPDU, it immediately returns to a normal spanning tree port type and moves through the regular transitions. In that case, BPDU Filtering is disabled on this port, and spanning tree resumes sending BPDUs on this port.

In addition, you can configure BPDU Filtering by the individual interface. When you explicitly configure BPDU Filtering on a port, that port does not send any BPDUs and drops all BPDUs that it receives. You can effectively override the global BPDU Filtering setting on individual ports by configuring the specific interface.

This BPDU Filtering command on the interface applies to the entire interface, whether the interface is trunking or not.

Caution Use care when configuring BPDU Filtering per interface. If you explicitly configure BPDU Filtering on a port that is not connected to a host, it can result in bridging loops because the port will ignore any BPDU that it receives and go to forwarding.

This table lists all the BPDU Filtering combinations.

Table 13: BPDU Filtering Configurations

BPDU Filtering Per Port Configuration	BPDU Filtering Global Configuration	STP Edge Port Configuration	BPDU Filtering State
Default [1]	Enable	Enable	Enable [2]
Default	Enable	Disable	Disable
Default	Disable	Not applicable	Disable
Disable	Not applicable	Not applicable	Disable
Enable	Not applicable	Not applicable	Enable

1 No explicit port configuration.
2 The port transmits at least 10 BPDUs. If this port receives any BPDUs, the port returns to the spanning tree normal port state and BPDU filtering is disabled.

Loop Guard

Loop Guard helps prevent bridging loops that could occur because of a unidirectional link failure on a point-to-point link.

An STP loop occurs when a blocking port in a redundant topology erroneously transitions to the forwarding state. Transitions are usually caused by a port in a physically redundant topology (not necessarily the blocking port) that stops receiving BPDUs.

When you enable Loop Guard globally, it is useful only in switched networks where devices are connected by point-to-point links. On a point-to-point link, a designated bridge cannot disappear unless it sends an inferior BPDU or brings the link down. However, you can enable Loop Guard on shared links per interface,

You can use Loop Guard to determine if a root port or an alternate/backup root port receives BPDUs. If the port that was previously receiving BPDUs is no longer receiving BPDUs, Loop Guard puts the port into an inconsistent state (blocking) until the port starts to receive BPDUs again. If such a port receives BPDUs again, the port—and link—is deemed viable again. The protocol removes the loop-inconsistent condition from the port, and the STP determines the port state because the recovery is automatic.

Loop Guard isolates the failure and allows STP to converge to a stable topology without the failed link or bridge. Disabling Loop Guard moves all loop-inconsistent ports to the listening state.

You can enable Loop Guard on a per-port basis. When you enable Loop Guard on a port, it is automatically applied to all of the active instances or VLANs to which that port belongs. When you disable Loop Guard, it is disabled for the specified ports.

Enabling Loop Guard on a root device has no effect but provides protection when a root device becomes a nonroot device.

Root Guard

When you enable Root Guard on a port, Root Guard does not allow that port to become a root port. If a received BPDU triggers an STP convergence that makes that designated port become a root port, that port is put into a root-inconsistent (blocked) state. After the port stops receiving superior BPDUs, the port is unblocked again. Through STP, the port moves to the forwarding state. Recovery is automatic.

When you enable Root Guard on an interface, this functionality applies to all VLANs to which that interface belongs.

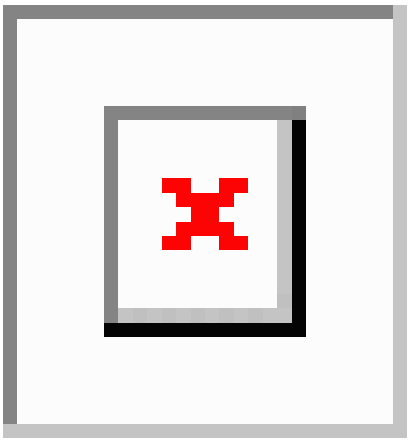
You can use Root Guard to enforce the root bridge placement in the network. Root Guard ensures that the port on which Root Guard is enabled is the designated port. Normally, root bridge ports are all designated ports, unless two or more of the ports of the root bridge are connected. If the bridge receives superior BPDUs on a Root Guard-enabled port, the bridge moves this port to a root-inconsistent STP state. In this way, Root Guard enforces the position of the root bridge.

You cannot configure Root Guard globally.

Applying STP Extension Features

Figure 19: Network with STP Extensions Correctly Deployed

We recommend that you configure the various STP extension features through your network as shown in this figure. Bridge Assurance is enabled on the entire network. You should enable either BPDU Guard or BPDU Filtering on the host interface.



PVST Simulation

MST interoperates with Rapid PVST+ with no need for user configuration. The PVST simulation feature enables this interoperability.

Note PVST simulation is enabled by default when you enable MST. By default, all interfaces on the device interoperate between MST and Rapid PVST+.

However, you may want to control the connection between MST and Rapid PVST+ to protect against accidentally connecting an MST-enabled port to a port enabled to run Rapid PVST+. Because Rapid PVST+ is the default STP mode, you may encounter many Rapid PVST+ connections.

Disabling Rapid PVST+ simulation, which can be done per port or globally for the entire device, moves the MST-enabled port to the blocking state once it detects it is connected to a Rapid PVST+-enabled port. This port remains in the inconsistent state until the port stops receiving Rapid PVST+/SSTP BPDUs, and then the port resumes the normal STP transition process.

The root bridge for all STP instances must all be in either the MST region or the Rapid PVST+ side. If the root bridge for all STP instances are not on one side or the other, the software moves the port into a PVST simulation-inconsistent state.

Note We recommend that you put the root bridge for all STP instances in the MST region.

High Availability for STP

The software supports high availability for STP. However, the statistics and timers are not restored when STP restarts. The timers start again and the statistics begin from 0.

Note See the *Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide*, for complete information on high-availability features.

Prerequisites for STP Extensions

STP has the following prerequisites:

- You must be logged onto the device.
- You must have STP configured already.

Guidelines and Limitations for Configuring STP Extensions

STP extensions have the following configuration guidelines and limitations:

- **show** commands with the **internal** keyword are not supported.
- Connect STP network ports only to switches.
- You should configure host ports as STP edge ports and not as network ports.

- If you enable STP network port types globally, ensure that you manually configure all ports connected to hosts as STP edge ports.
- You should configure all access and trunk ports connected to Layer 2 hosts as edge ports.
- Bridge Assurance runs only on point-to-point spanning tree network ports. You must configure each side of the link for this feature.
- We recommend that you enable Bridge Assurance throughout your network.
- We recommend that you enable BPDU Guard on all edge ports.
- Enabling Loop Guard globally works only on point-to-point links.
- Enabling Loop Guard per interface works on both shared and point-to-point links.
- Root Guard forces a port to always be a designated port; it does not allow a port to become a root port. Loop Guard is effective only if the port is a root port or an alternate port. You cannot enable Loop Guard and Root Guard on a port at the same time.
- Loop Guard has no effect on a disabled spanning tree instance or a VLAN.
- Spanning tree always chooses the first operational port in the channel to send the BPDUs. If that link becomes unidirectional, Loop Guard blocks the channel, even if other links in the channel are functioning properly.
- If you group together a set of ports that are already blocked by Loop Guard to form a channel, spanning tree loses all the state information for those ports and the new channel port may obtain the forwarding state with a designated role.
- If a channel is blocked by Loop Guard and the channel members go back to an individual link status, spanning tree loses all the state information. The individual physical ports may obtain the forwarding state with the designated role, even if one or more of the links that formed the channel are unidirectional.

Note You can enable UniDirectional Link Detection (UDLD) aggressive mode to isolate the link failure. A loop may occur until UDLD detects the failure, but Loop Guard will not be able to detect it. See the *Cisco NX-OSSeries NX-OS Interfaces Configuration Guide*, for information on UDLD.

- You should enable Loop Guard globally on a switch network with physical loops.
- You should enable Root Guard on ports that connect to network devices that are not under direct administrative control.
- After the maximum MAC learning limit , all incoming packets are not learnt in the MAC table and are forwarded based on the destination MAC.
- Beginning with Cisco NX-OS Release 10.5(3)F, STP is supported on the Cisco Nexus 93C64E-SG2-Q switch.

Default Settings for STP Extensions

This table lists the default settings for STP extensions.

Table 14: Default STP Extension Parameters

Parameters	Default
Port type	Normal
Bridge Assurance	Enabled (on STP network ports only)
Global BPDU Guard	Disabled
BPDU Guard per interface	Disabled
Global BPDU Filtering	Disabled
BPDU Filtering per interface	Disabled
Global Loop Guard	Disabled
Loop Guard per interface	Disabled
Root Guard per interface	Disabled
PVST simulation	Enabled

Configuring STP Extensions Steps

Note If you are familiar with the Cisco IOS CLI, be aware that the Cisco NX-OS commands for this feature might differ from the Cisco IOS commands that you would use.

You can enable Loop Guard per interface on either shared or point-to-point links.

Configuring Spanning Tree Port Types Globally

The spanning tree port type designation depends on the device the port is connected to, as follows:

- Edge—Edge ports are connected to Layer 2 hosts and are access ports.
- Network—Network ports are connected only to Layer 2 switches or bridges and can be either access or trunk ports.
- Normal—Normal ports are neither edge ports nor network ports; they are normal spanning tree ports. These ports can be connected to any device.

You can configure the port type either globally or per interface. By default, the spanning tree port type is normal.

Before you begin

Before you configure the spanning port type, you should do the following:

- Ensure that STP is configured.
- Ensure that you are configuring the ports correctly as to the device to which the port is connected.

SUMMARY STEPS

1. **config t**
2. **spanning-tree port type edge default** or **spanning-tree port type network default**
3. **exit**
4. (Optional) **show spanning-tree summary**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**spanning-tree port type edge default** or **spanning-tree port type network default** **Example:** `switch(config)# spanning-tree port type edge default`	• **spanning-tree port type edge default** Configures all access ports connected to Layer 2 hosts as edge ports. Edge ports immediately transition to the forwarding state without passing through the blocking or learning state at linkup. By default, spanning tree ports are normal port types. • **spanning-tree port type network default** Configures all interfaces connected to Layer 2 switches and bridges as spanning tree network ports. If you enable Bridge Assurance, it automatically runs on network ports. By default, spanning tree ports are normal port types. **Note** If you configure interfaces connected to Layer 2 hosts as network ports, those ports automatically move into the blocking state.

	Command or Action	Purpose

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

Step 4	(Optional) **show spanning-tree summary** **Example:** `switch# show spanning-tree summary`	Displays the STP configuration including STP port types if configured.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure all access ports connected to Layer 2 hosts as spanning tree edge ports:

```
switch# config t
switch(config)# spanning-tree port type edge default
switch(config)# exit
switch#
```

This example shows how to configure all ports connected to Layer 2 switches or bridges as spanning tree network ports:

```
switch# config t
switch(config)# spanning-tree port type network default
switch(config)# exit
switch#
```

Configuring Spanning Tree Edge Ports on Specified Interfaces

You can configure spanning tree edge ports on specified interfaces. Interfaces configured as spanning tree edge ports immediately transition to the forwarding state, without passing through the blocking or learning states, on linkup.

This command has four states:

- **spanning-tree port type edge**—This command explicitly enables edge behavior on the access port.
- **spanning-tree port type edge trunk**—This command explicitly enables edge behavior on the trunk port.

Note If the **spanning-tree port type edge trunk** configuration is applied and a BPDU is received on the port, then portfast feature is disabled. As a result, the port does not function as an edge port.

Note If you enter the **spanning-tree port type edge trunk** command, the port is configured as an edge port even in the access mode.

- **spanning-tree port type normal**—This command explicitly configures the port as a normal spanning tree port and the immediate transition to the forwarding state is not enabled.
- **no spanning-tree port type**—This command implicitly enables edge behavior if you define the **spanning-tree port type edge default** command in global configuration mode. If you do not configure the edge ports globally, the **no spanning-tree port type** command is equivalent to the **spanning-tree port type normal** command.

Before you begin

Before you configure the spanning port type, you should do the following:

- Ensure that STP is configured.
- Ensure that you are configuring the ports correctly as to the device to which the port is connected.

SUMMARY STEPS

1. **config t**
2. **interface** *type slot/port*
3. **spanning-tree port type edge**
4. **exit**
5. (Optional) **show spanning-tree interface** *type slot/port* **ethernet** *x/y*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)#`
		Specifies the interface to configure, and enters the interface configuration mode.

Step 3	**spanning-tree port type edge** **Example:**	Configures the specified access interfaces to be spanning edge ports. Edge ports immediately transition to the forwarding state without passing through the blocking or

	Command or Action	Purpose
	`switch(config-if)# spanning-tree port type edge`	learning state at linkup. By default, spanning tree ports are normal port types.

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

Step 5	(Optional) **show spanning-tree interface** *type slot/port* **ethernet** *x/y* **Example:** `switch# show spanning-tree ethernet 1/4`	Displays the STP configuration including the STP port type if configured.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the Ethernet access interface 1/4 to be a spanning tree edge port:

```
switch# config t
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree port type edge
switch(config-if)# exit
switch(config)#
```

Configuring Spanning Tree Network Ports on Specified Interfaces

You can configure spanning tree network ports on specified interfaces.

Bridge Assurance runs only on spanning tree network ports.

This command has three states:

- **spanning-tree port type network**—This command explicitly configures the port as a network port. If you enable Bridge Assurance globally, it automatically runs on a spanning tree network port.
- **spanning-tree port type normal** —This command explicitly configures the port as a normal spanning tree port and Bridge Assurance cannot run on this interface.
- **no spanning-tree port type**—This command implicitly enables the port as a spanning tree network port if you define the **spanning-tree port type network default** command in global configuration mode. If you enable Bridge Assurance globally, it automatically runs on this port.

Note A port connected to Layer 2 hosts that is configured as a network ports automatically moves into the blocking state.

Before you begin

Before you configure the spanning port type, you should do the following:

- Ensure that STP is configured.
- Ensure that you are configuring the ports correctly as to the device to which the port is connected.

SUMMARY STEPS

1. **config t**
2. **interface** *type slot/port*
3. **spanning-tree port type network**
4. **exit**
5. (Optional) **show spanning-tree interface** *type slot/port*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)#`
		Specifies the interface to configure, and enters the interface configuration mode.

Step 3	**spanning-tree port type network** **Example:** `switch(config-if)# spanning-tree port type network`	Configures the specified interfaces to be spanning network ports. If you enable Bridge Assurance, it automatically runs on network ports. By default, spanning tree ports are normal port types.

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

	Command or Action	Purpose
Step 5	(Optional) **show spanning-tree interface** *type slot/port* **Example:** `switch# show spanning-tree interface ethernet 1/4`	Displays the STP configuration including the STP port type if configured.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to configure the Ethernet interface 1/4 to be a spanning tree network port:

```
switch# config t
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree port type network
switch(config-if)# exit
switch(config)#
```

Enabling BPDU Guard Globally

You can enable BPDU Guard globally by default. In this condition, the system shuts down an edge port that receives a BPDU.

Note We recommend that you enable BPDU Guard on all edge ports.

Before you begin

Before you configure the spanning port type, you should do the following:

- Ensure that STP is configured.
- Ensure that you are configuring the ports correctly as to the device to which the port is connected.

SUMMARY STEPS

1. **config t**
2. **spanning-tree port type edge bpduguard default**
3. **exit**
4. (Optional) **show spanning-tree summary**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**spanning-tree port type edge bpduguard default** **Example:** `switch(config)# spanning-tree port type edge bpduguard default`	Enables BPDU Guard by default on all spanning tree edge ports. By default, global BPDU Guard is disabled.

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

Step 4	(Optional) **show spanning-tree summary** **Example:** `switch# show spanning-tree summary`	Displays summary STP information.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to enable BPDU Guard on all spanning tree edge ports:

```
switch# config t
switch(config)# spanning-tree port type edge bpduguard default
switch(config)# exit
switch#
```

Enabling BPDU Guard on Specified Interfaces

You can enable BPDU Guard on specified interfaces. Enabling BPDU Guard shuts down the port if it receives a BPDU.

You can configure BPDU Guard on specified interfaces as follows:

- **spanning-tree bpduguard enable** —Unconditionally enables BPDU Guard on the interface.
- **spanning-tree bpduguard disable** —Unconditionally disables BPDU Guard on the interface.

- **no spanning-tree bpduguard** —Enables BPDU Guard on the interface if it is an operational edge port and if the **spanning-tree port type edge bpduguard default** command is configured.

Before you begin

Before you configure this feature, you should do the following:

- Ensure that STP is configured.

SUMMARY STEPS

1. **config t**
2. **interface** *type slot/port*
3. **spanning-tree bpduguard** {**enable** | **disable**} or **no spanning-tree bpduguard**
4. **exit**
5. (Optional) **show spanning-tree interface** *type slot/port* **detail**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)#`
		Specifies the interface to configure, and enters the interface configuration mode.

Step 3	**spanning-tree bpduguard** {**enable** | **disable**} or **no spanning-tree bpduguard** **Example:** `switch(config-if)# spanning-tree bpduguard enable`	• **spanning-tree bpduguard** {**enable** | **disable**} Enables or disables BPDU Guard for the specified spanning tree edge interface. By default, BPDU Guard is disabled on the interfaces. • **no spanning-tree bpduguard** Falls back to the default BPDU Guard global setting that you set for the interfaces by entering the **spanning-tree port type edge bpduguard default** command.
Step 4	**exit** **Example:**	Exits interface mode.

	Command or Action	Purpose
	`switch(config-if)# exit` `switch(config)#`	
Step 5	(Optional) **show spanning-tree interface** *type slot/port* **detail** **Example:** `switch# show spanning-tree interface ethernet detail`	Displays summary STP information.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to explicitly enable BPDU Guard on the Ethernet edge port 1/4:

```
switch# config t
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree bpduguard enable
switch(config-if)# exit
switch(config)#
```

Enabling BPDU Filtering Globally

You can enable BPDU Filtering globally by default on spanning tree edge ports.

If an edge port with BPDU Filtering enabled receives a BPDU, it loses its operation status as edge port and resumes the regular STP transitions. However, this port maintains its configuration as an edge port.

Caution Be careful when using this command. Using this command incorrectly can cause bridging loops.

Before you begin

Before you configure this feature, you should do the following:

- Ensure that STP is configured.
- Ensure that you have configured some spanning tree edge ports.

Note When enabled globally, BPDU Filtering is applied only on ports that are operational edge ports. Ports send a few BPDUs at linkup before they effectively filter outbound BPDUs. If a BPDU is received on an edge port, it immediately loses its operational edge port status and BPDU Filtering is disabled.

SUMMARY STEPS

1. **config t**
2. **spanning-tree port type edge bpdufilter default**
3. **exit**
4. (Optional) **show spanning-tree summary**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**spanning-tree port type edge bpdufilter default** **Example:** `switch(config)# spanning-tree port type edge bpdufilter default`	Enables BPDU Filtering by default on all operational spanning tree edge ports. Global BPDU Filtering is disabled by default.

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

Step 4	(Optional) **show spanning-tree summary** **Example:** `switch# show spanning-tree summary`	Displays summary STP information.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to enable BPDU Filtering on all operational spanning tree edge ports:

```
switch# config t
switch(config)# spanning-tree port type edge bpdufilter default
switch(config)# exit
switch#
```

Enabling BPDU Filtering on Specified Interfaces

You can apply BPDU Filtering to specified interfaces. When enabled on an interface, that interface does not send any BPDUs and drops all BPDUs that it receives. This BPDU Filtering functionality applies to the entire interface, whether trunking or not.

Caution Be careful when you enter the **spanning-tree bpdufilter enable** command on specified interfaces. Explicitly configuring BPDU Filtering on a port that is not connected to a host can result in bridging loops because the port will ignore any BPDU that it receives and go to forwarding.

You can enter this command to override the port configuration on specified interfaces.

This command has three states:

- **spanning-tree bpdufilter enable**—Unconditionally enables BPDU Filtering on the interface.
- **spanning-tree bpdufilter disable**—Unconditionally disables BPDU Filtering on the interface.
- **no spanning-tree bpdufilter** ——Enables BPDU Filtering on the interface if the interface is in operational edge port and if you configure the **spanning-tree port type edge bpdufilter default** command.

Before you begin

Before you configure this feature, you should do the following:

- Ensure that STP is configured.

Note When you enable BPDU Filtering locally on a port, this feature prevents the device from receiving or sending BPDUs on this port.

SUMMARY STEPS

1. **config t**
2. **interface** *type slot/port*
3. **spanning-tree bpdufilter** {**enable** | **disable**} or **no spanning-tree bpdufilter**
4. **exit**
5. (Optional) **show spanning-tree summary**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)#`
		Specifies the interface to configure, and enters the interface configuration mode.

Step 3	**spanning-tree bpdufilter** {**enable** | **disable**} or **no spanning-tree bpdufilter** **Example:** `switch(config-if)# spanning-tree bpdufilter enable`	• **spanning-tree bpdufilter** {**enable** | **disable**} Enables or disables BPDU Filtering for the specified spanning tree edge interface. By default, BPDU Filtering is disabled. • **no spanning-tree bpdufilter** Enables BPDU Filtering on the interface if the interface is an operational spanning tree edge port and if you enter the **spanning-tree port type edge bpdufilter default** command.

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

Step 5	(Optional) **show spanning-tree summary** **Example:** `switch# show spanning-tree summary`	Displays summary STP information.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to explicitly enable BPDU Filtering on the Ethernet spanning tree edge port 1/4:

```
switch# config t
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree bpdufilter enable
```

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

Enabling Loop Guard Globally

You can enable Loop Guard globally by default on all point-to-point spanning tree normal and network ports. Loop Guard does not run on edge ports.

Loop Guard provides additional security in the bridge network. Loop Guard prevents alternate or root ports from becoming the designated port because of a failure that could lead to a unidirectional link.

Note Entering the Loop Guard command for the specified interface overrides the global Loop Guard command.

Before you begin

Before you configure this feature, you should do the following:

- Ensure that STP is configured.
- Ensure that you have spanning tree normal ports or have configured some network ports.

SUMMARY STEPS

1. **config t**
2. **spanning-tree loopguard default**
3. **exit**
4. (Optional) **show spanning-tree summary**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**spanning-tree loopguard default** **Example:** `switch(config)# spanning-tree loopguard default`	Enables Loop Guard by default on all spanning tree normal and network ports. By default, global Loop Guard is disabled.

	Command or Action	Purpose

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

Step 4	(Optional) **show spanning-tree summary** **Example:** `switch# show spanning-tree summary`	Displays summary STP information.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to enable Loop Guard on all spanning tree normal or network ports:

```
switch# config t
switch(config)# spanning-tree loopguard default
switch(config)# exit
switch#
```

Enabling Loop Guard or Root Guard on Specified Interfaces

Note You can run Loop Guard on spanning tree normal or network ports. You can run Root Guard on all spanning tree ports: normal, edge, or network.

You can enable either Loop Guard or Root Guard on specified interfaces.

Enabling Root Guard on a port means that port cannot become a root port, and Loop Guard prevents alternate or root ports from becoming the designated port because of a failure that could lead to a unidirectional link.

Both Loop Guard and Root Guard enabled on an interface apply to all VLANs to which that interface belongs.

Note Entering the Loop Guard command for the specified interface overrides the global Loop Guard command.

Before you begin

Before you configure this feature, you should do the following:

- Ensure that STP is configured.
- Ensure that you are configuring Loop Guard on spanning tree normal or network ports.

SUMMARY STEPS

1. **config t**
2. **interface** *type slot/port*
3. **spanning-tree guard** {**loop** | **root** | **none**}
4. **exit**
5. **interface** *type slot/port*
6. **spanning-tree guard** {**loop** | **root** | **none**}
7. **exit**
8. (Optional) **show spanning-tree interface** *type slot/port* **detail**
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/4`
	`switch(config-if)#`
		Specifies the interface to configure, and enters the interface configuration mode.

Step 3	**spanning-tree guard** {**loop**	**root**	**none**} **Example:** `switch(config-if)# spanning-tree guard loop`	Enables or disables either Loop Guard or Root Guard for the specified interface. By default, Root Guard is disabled by default, and Loop Guard on specified ports is also disabled. **Note** Loop Guard runs only on spanning tree normal and network interfaces. This example shows Loop Guard is enabled on the specified interface.

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

Step 5	**interface** *type slot/port*
	Example:
	`switch(config)# interface ethernet 1/10`
	`switch(config-if)#`
		Specifies the interface to configure, and enters the interface configuration mode.

	Command or Action	Purpose
Step 6	**spanning-tree guard** {**loop** | **root** | **none**} **Example:** `switch(config-if)# spanning-tree guard root`	Enables or disables either Loop Guard or Root Guard for the specified interface. By default, Root Guard is disabled by default, and Loop Guard on specified ports is also disabled. The example shows Root Guard is enabled on a different interface.

Step 7	**exit**
	Example:
	`switch(config-if)# exit`
	`switch(config)#`
		Exits interface mode.

Step 8	(Optional) **show spanning-tree interface** *type slot/port* **detail** **Example:** `switch# show spanning-tree interface ethernet 1/4 detail`	Displays summary STP information.
Step 9	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to enable Root Guard on Ethernet port 1/4:

```
switch# config t
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree guard root
switch(config-if)# exit
switch(config)#
```

Configuring PVST Simulation Globally-CLI Version

Note PVST simulation is enabled by default. By default, all interfaces on the device interoperate between MST and Rapid PVST+.

MST interoperates with Rapid PVST+. However, to prevent an accidental connection to a device that does not run MST as the default STP mode, you may want to disable this automatic feature. If you disable PVST simulation, the MST-enabled port moves to the blocking state once it detects it is connected to a Rapid PVST+-enabled port. This port remains in the inconsistent state until the port stops receiving BPDUs, and then the port resumes the normal STP transition process.

You can block this automatic feature either globally or per port. You can enter the global command and change the PVST simulation setting for the entire device while you are in interface command mode.

SUMMARY STEPS

1. **config t**
2. **no spanning-tree mst simulate pvst global**
3. **exit**
4. (Optional) **show spanning-tree summary**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**no spanning-tree mst simulate pvst global** **Example:** `switch(config)# no spanning-tree mst simulate pvst global`	Disables all interfaces on the switch from automatically interoperating with a connected device that is running in Rapid PVST+ mode. The default for this is enabled; by default, all interfaces on the device operate between Rapid PVST+ and MST.

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

Step 4	(Optional) **show spanning-tree summary** **Example:** `switch# show spanning-tree summary`	Displays detailed STP information.
Step 5	(Optional) **copy running-config startup-config** **Example:** `switch# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to prevent the device from automatically interoperating with a connecting device that is running Rapid PVST+:

```
switch# config t
switch(config)# no spanning-tree mst simulate pvst global
switch(config)# exit
switch#
```

Configuring PVST Simulation Per Port

Note PVST simulation is enabled by default. By default, all interfaces on the device interoperate between MST and Rapid PVST+.

You can configure PVST simulation only when you are running MST on the device (Rapid PVST+ is the default STP mode). MST interoperates with Rapid PVST+. However, to prevent an accidental connection to a device that does not run MST as the default STP mode, you may want to disable this automatic feature. If you disable PVST simulation, the MST-enabled port moves to the blocking state once it detects that it is connected to a Rapid PVST+-enabled port. This port remains in the inconsistent state until the port stops receiving Rapid PVST+ BPDUs, and then the port resumes the normal STP transition process.

You can block this automatic feature either globally or per port.

SUMMARY STEPS

1. **config t**
2. **interface** {{*type slot/port*} |{**port-channel** *number*}}
3. **spanning-tree mst simulate pvst disable** or **spanning-tree mst simulate pvst** or **no spanning-tree mst simulate pvst**
4. **exit**
5. (Optional) **show spanning-tree interface** *type slot/port* **detail**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose

Step 1	**config t**
	Example:
	`switch# config t`
	`switch(config)#`
		Enters configuration mode.

Step 2	**interface** {{*type slot/port*} |{**port-channel** *number*}}
	Example:
	`switch(config)# interface ethernet 3/1`
	`switch(config-if)#`
		Specifies an interface to configure, and enters interface configuration mode.

Step 3	**spanning-tree mst simulate pvst disable** or **spanning-tree mst simulate pvst** or **no spanning-tree mst simulate pvst** **Example:** `switch(config-if)# spanning-tree mst simulate pvst`	• **spanning-tree mst simulate pvst disable** Disables specified interfaces from automatically interoperating with a connected device that is running in Rapid PVST+ mode. By default, all interfaces on the device operate between Rapid PVST+ and MST.

	Command or Action	Purpose
		• **spanning-tree mst simulate pvst** Reenables seamless operation between MST and Rapid PVST+ on specified interfaces. • **no spanning-tree mst simulate pvst** Sets the interface to the device-wide MST and Rapid PVST+ interoperation that you configured using the **spanning-tree mst simulate pvst global** command.

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

Step 5	(Optional) **show spanning-tree interface** *type slot/port* **detail** **Example:** `switch# show spanning-tree interface ethernet 3/1 detail`	Displays detailed STP information.
Step 6	(Optional) **copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	Copies the running configuration to the startup configuration.

Example

This example shows how to prevent the specified interfaces from automatically interoperating with a connecting device that is not running MST:

```
switch(config-if)# spanning-tree mst simulate pvst
switch(config-if)#
```

Verifying the STP Extension Configuration

To display the configuration information for the STP extensions, perform one of the following tasks:

Command	Purpose
show running-config spanning-tree [**all**]	Displays information about STP.
show spanning-tree summary	Displays summary information on STP.
show spanning-tree mst *instance-id* **interface** {**ethernet** *slot/port* | **port-channel** *channel-number*} [**detail**]	Displays MST information for the specified interface and instance.

Configuration Examples for STP Extension

The following example shows how to configure the STP extensions:

```
switch# configure terminal
switch(config)# spanning-tree port type network default
switch(config)# spanning-tree port type edge bpduguard default
switch(config)# spanning-tree port type edge bpdufilter default

switch(config)# interface ethernet 1/1
switch(config-if)# spanning-tree port type edge
switch(config-if)# exit

switch(config)# interface ethernet 1/2
switch(config-if)# spanning-tree port type edge
switch(config-if)# exit
switch(config)#
```

Additional References for STP Extensions -- CLI Version

Related Documents

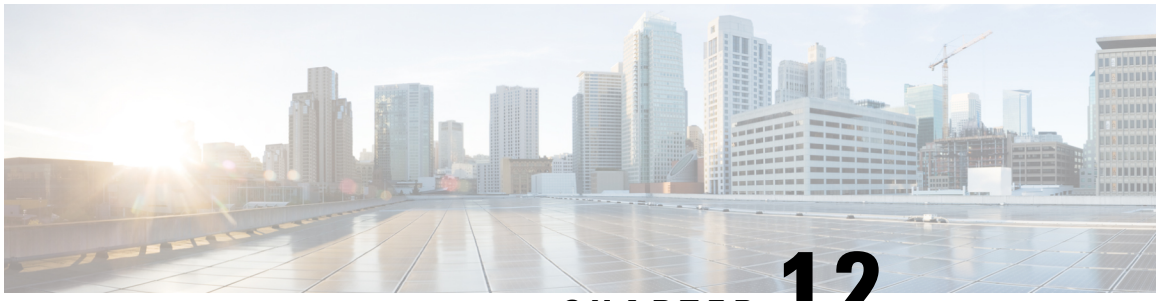
Related Topic	Document Title
Layer 2 interfaces	*Cisco Nexus 9000 Series NX-OS Interfaces Configuration Guide*
NX-OS fundamentals	*Cisco Nexus 9000 Series NX-OS Fundamentals Configuration Guide*
High availability	*Cisco Nexus 9000 Series NX-OS High Availability and Redundancy Guide*
System management	*Cisco Nexus 9000 Series NX-OS System Management Configuration Guide*

Standards

Standards	Title
IEEE 802.1Q-2006 (formerly known as IEEE 802.1s), IEEE 802.1D-2004 (formerly known as IEEE 802.1w), IEEE 802.1D, IEEE 802.1t	—

MIBs

MIBs	MIBs Link
• CISCO-STP-EXTENSION-MIB • BRIDGE-MIB	To locate and download MIBs, go to the following URL: https://cisco.github.io/cisco-mibs/supportlists/nexus9000/Nexus9000MIBSupportList.html



CHAPTER 12

Configuring Reflective Relay for Layer 2 Switching

- About Reflective Relay 802.1Qbg, on page 205
- Guidelines and Limitations for Reflective Relay, on page 205
- Configuring Reflective Relay Using the NX-OS CLI, on page 206

About Reflective Relay 802.1Qbg

Reflective relay is a tagless approach of IEEE standard 802.1Qbg. It forwards all traffic to an external switch that applies policy and sends the traffic back to the destination or target VM on the server as needed. There is no local switching. For broadcast or multicast traffic, reflective relay provides packet replication to each VM locally on the server.

Reflective relay leverages the external switch for switching features and management capabilities, freeing server resources to support the VMs. Reflective relay applies the policies you configure on the Cisco Nexus switches to traffic between the VMs on the same server.

You can enable reflective relay to turn back traffic out of the same port it came in on. You can enable reflective relay on a Layer 2 physical port or port-channel interface policy using the NX-OS CLI. This feature is disabled by default.

The term Virtual Ethernet Port Aggregator (VEPA) is also used to describe 802.1Qbg functionality.

Reflective Relay Support

Nexus Switches introduces support for Reflective relay in these releases:

Guidelines and Limitations for Reflective Relay

Reflective relay has these configuration guidelines or limitations:

- IEEE standard 802.1Qbg tagless approach, known as reflective relay.
- Physical domains—virtual domains are not supported.

- Physical ports and port channels—Does not support Cisco Fabric Extender (FEX) and blade servers. If reflective relay is enabled on an unsupported interface, a fault is raised, and the last valid configuration is retained. Disabling reflective relay on the port clears the fault.
- ARP suppression must be disabled before using the reflective relay feature.

Configuring Reflective Relay Using the NX-OS CLI

Reflective relay is disabled by default; however, you can enable it on a port or port channel as a Layer 2 interface policy on the switch. In the NX-OS CLI, you can use a template to enable reflective relay on multiple ports or you can enable it on individual ports.

Procedure

Step 1 **configure terminal**

Example:

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

Enters global configuration mode.

Step 2 **interface ethernet 1/2**

Example:

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

Enables the port.

Step 3 **switchport virtual-ethernet-bridge**

Example:

```
switch(config-if)# switchport virtual-ethernet-bridge
switch(config-if)#
```

Configures the Layer 2 port as a host port for the reflective relay feature.

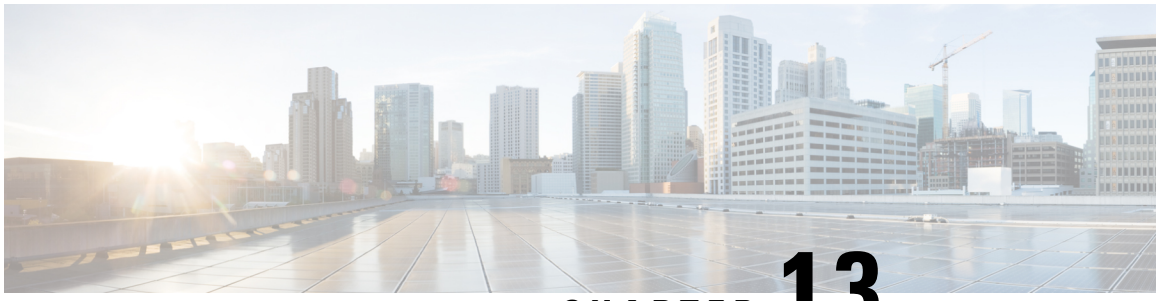
Step 4 **[no] switchport virtual-ethernet-bridge**

Example:

```
switch(config-if)# no switchport virtual-ethernet-bridge
```

Enables the reflective relay feature.

Note
The reflective relay feature is only supported on access or trunk ports.



CHAPTER 13

Configuring Link-State Tracking

This chapter describes how to configure Link-State Tracking on the Cisco NX-OS device.

This chapter includes the following sections:

- Link-State Tracking, on page 207
- How Link-State Tracking Works, on page 207
- Guidelines and Limitation for Link-State Tracking, on page 208
- Configure Link-State Tracking, on page 209
- Monitor the Link-State Tracking, on page 210
- Link-State Tracking Configuration Examples, on page 212

Link-State Tracking

Link-State Tracking is a mechanism that binds multiple interface link states to ensure redundancy and network stability. It is particularly useful in environments with server NIC adapters in a primary-secondary relationship, allowing seamless transition to secondary links if the primary fails.

- **Uplink and Downlink Interfaces**: Interfaces are categorized as uplink (connected to distribution switches or routers) or downlink (connected to servers or end devices).
- **Failover Mechanism**: If all uplink interfaces in a group fail, the downlink interfaces are automatically disabled, triggering a failover to secondary links.

How Link-State Tracking Works

Summary

Link-state tracking involves grouping interfaces into uplink and downlink categories and monitoring their connectivity to maintain network availability.

This process ensures that network connectivity is maintained by automatically redirecting traffic and administratively controlling interface states in response to link failures.

Workflow

These stages describe how link-state tracking works.

1. Interfaces are grouped into link-state groups, categorized as either uplink or downlink interfaces.
   - Uplink interfaces typically connect to distribution switches or routers, while downlink interfaces connect to end devices like servers.
2. The system continuously monitors the link state of uplink interfaces.
   - If at least one uplink interface is active, the downlink interfaces remain in the "link-up" state.
   - If all uplink interfaces fail, downlink interfaces will transition to the **trackedPortDown** state. This indicates that the interface has been administratively disabled by the link-state tracking feature.
3. When uplink connectivity is lost, the system initiates a failover to redirect traffic to secondary interfaces.
   - This ensures that the network remains operational despite link failures.
4. Downlink interfaces can be recovered by either removing the failed uplink interfaces from the group or disabling the link-state group entirely.
   - After uplink connectivity is restored, the downlink bringup delay timer stabilizes the transition of downlink interfaces to an active state. The default downlink bringup delay timer value is 60 seconds. The maximum configurable value is 600 seconds.

In a typical setup, servers connect to two switches (primary and secondary) using uplink and downlink interfaces.

If the primary switch loses connectivity, the system automatically switches traffic to the secondary switch, maintaining network availability

Guidelines and Limitation for Link-State Tracking

- You can add an interface to only one link-state group at a time. You cannot include it in both an upstream and downstream group.

Note If an interface is configured as both an uplink and a downlink in the same group, an error will be generated.

- A maximum of 64 link-state groups can be created. Group names must not exceed 64 characters.
- Currently, only physical and breakout interfaces are supported. SVIs and port channels are not supported.
- Do not enable link-state tracking on individual interfaces that are part of a downlink EtherChannel interface.
- Duplicate entries for uplinks or downlinks within a group are not allowed. Each interface must be unique within its group.
- Subinterfaces are not supported for link-state tracking.

Configure Link-State Tracking

This section provides a step-by-step guide to configure Link-State Tracking on Cisco Nexus 9000 Series NX-OS devices.

Before you begin

Ensure that all interfaces intended for use in link-state groups are operational and not administratively shut down.

Procedure

Step 1 Use the **configure terminal** command to enter global configuration mode.

Example:

```
switch# configure terminal
```

Step 2 Use the **link state group** *group-name* command to create a link-state group and enable link-state tracking.

Example:

```
switch(config)# link state group <group-name>
```

To remove the entire link-state group configuration:

```
switch(config)# no link state group <group-name>
```

Replace <*group-name*> with the desired group name (e.g., group-1).

```
switch(config)# link state group group-1
            switch(config-link-state)#
```

Note
Group names can be a maximum of 64 characters long.

Step 3 Use the **downlink bringup delay** *seconds* command to configure the delay before downstream interfaces are brought back up after uplink connectivity is restored.

Example:

```
switch(config-link-state)# downlink bringup delay ?
            <1 - 600> Timer in seconds
```

To remove the downlink bringup delay timer configuration:

```
switch(config-link-state)# no downlink bringup delay
```

The default value is 60 seconds. The configurable range is 1 to 600 seconds.

```
switch(config-link-state)# downlink bringup delay 60
```

Step 4 Use the **interface** *interface-id* command to specify the interface or range of interfaces to configure.

Example:

```
switch(config)# interface <interface-id>
```

Replace <*interface-id*> with the specific interface or range of interfaces to configure (e.g., Ethernet1/1).

```
switch(config)# interface Ethernet1/1
            switch(config-if)#
```

Step 5 Use the **link state group** *group-name* **type** {uplink | downlink} command to assign the interface to a link-state group as an uplink or downlink.

Example:

```
switch(config-if)# link state group <group-name> type {uplink | downlink}
```

To remove an interface from a link-state group:

```
switch(config-if)# no link state group <group-name> type {uplink | downlink}
```

Replace <*group-name*> with the group name created earlier.

Use *uplink* for interfaces connected to distribution switches or routers.

```
switch(config-if)# link state group group-1 type uplink
```

Use *downlink* for interfaces connected to end devices like servers.

```
switch(config-if)# link state group group-1 type downlink
```

Note
A single link-state group can have a maximum of 64 uplink interfaces and 64 downlink interfaces, for a total of 128 interfaces.

Step 6 Use the **end** command to exit configuration mode.

Example:

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

Monitor the Link-State Tracking

To monitor and verify the configuration and status of Link-State Tracking, follow these steps:

Procedure

Step 1 Use the **show link state group all** command to display detailed information about the link-state group.

Example:

```
switch# show link state group all
Link State Group: group-1
Status: Up
Uplink Interfaces: Eth1/1
Downlink Interfaces: Eth1/2
Number of times disabled: 2
Last disabled: Sat Mar 1 17:10:26 2025
Downlink Delay Timer: 60
```

Displays detailed information about the link-state group, including:

- Group status (Up/Down/Pending)
- Uplink and Downlink interfaces
- Number of times the group was disabled

- Last disabled timestamp
- Downlink bringup delay timer

`show link state group <name>` displays detailed information about a specific group.

Step 2 Use the **show run int eth <port>** command to display the running configuration of a specific interface and verify if it is part of an uplink or downlink group.

Example:

```
switch# show run int eth 1/1
!Command: show running-config interface Ethernet1/1
!Running configuration last done at: Sat Mar 1 17:10:26 2025
!Time: Sat Mar 1 17:15:42 2025
version 10.4(4o)
Bios: version 01.07
interface Ethernet1/1
  link state group group-1 type uplink
  no shutdown
```

Step 3 Use the **show system internal ethpm info interface eth <port>** command to display detailed internal information about a specific interface, including its administrative and operational states, link state group membership, and other diagnostic data to verify its role as an uplink or downlink interface.

Example:

```
switch# show system internal ethpm info interface eth1/1
Ethernet1/1 - if_index: 0x1A000000
Backplane MAC address: d0:09:c8:63:65:90
Router MAC address:    d0:09:c8:63:65:8f

Admin Config Information:
  state(up), mode(trunk), speed(1 Gbps), duplex(Full), medium_db(0), fec(auto),
port-type-fabric(not-fabric) port-type-external(not-external) dfe-adaptive-tuning (1)
  layer(L2), dce-mode(edge), description(LACP Link to Switch-B), transmit-reset-skip(0)
  mac-up-timer(0), max-bringup-timer(0), active-jitter-mgmt(0), media-type(none) link loopback(off)
  auto neg(on), auto mdix(on), beacon(off), num_of_si(0)
  config auto neg(on) openflow(0) switchport_isolated(0)
  medium(broadcast), snmp trap(on), MTU(1500),
  flowcontrol rx(off) tx(off), link debounce(100),link debounce(link-up 0)
  storm-control bcast:100.00% mcast:100.00% ucast:100.00%
  storm-control bcast level1:100.00% bcast level2:100.00%  mcast level1:100.00% mcast level2:100.00%
 ucast level1:100.00% uncast level2:100.00%
  storm-control pps bcast:4294967295 mcast:4294967295 ucast:4294967295
  storm-control mcast:200 ucast:200
   span mode(0 - not a span-destination)
  trans mode(not a trans-port)
  Flexlink mode(disabled), Active Interface(none), Backup Interface(none), cfg_instance(0)
  delay(1), bw(1000000), rate-mode(dedicated)
  eee(n/a), eee_lpi(Normal), eee_latency(Constant)
  fabricpath enforce (DCE Core)(0)
  load interval [1-3]: 30, 300, 0 (sec).
  pkt timestamping: state =0, ingress_source_id=0, egress_source_id=0 .
  Link-sync group id 0
  Priority Extend -1
  Ethertype 0x8100
  Slowdrain Congestion : mode core timeout[500], mode edge [500]
  Slowdrain Pause : mode core enabled [n] timeout[0]
  Slowdrain Pause : mode edge enabled [n] timeout[500]
  MAC-addr IPv6-extract:0
  Buffer Boost:0 Speed Group:0, MAC Learn: Enabled
  shut lan (disabled)
  virtual-ethernet-bridge (Disabled)
```

```
  link state group name: group-1
hash-mode gtp-inner-v4 (Disabled)
```

Displays detailed information about:

- Internal port status and administrative configurations.
- Information about the link state group, including whether the group name has been cleared.
- Ensures that stale group information is not displayed by resetting the group name in the port configuration.

Step 4 Use the **show system internal dme running-config all dn sys/linkstate** command to display internal details about the link-state group configuration.

Example:

```
show system internal dme running-config all dn sys/linkstate
{
    "linkStateGroup": {
        "attributes": {
            "groupName": "group-1",
            "uplinkIntfList": ["eth1/1"],
            "downlinkIntfList": ["eth1/2"],
            "status": "up",
            "numDisabled": 2,
            "lastDisabledTimestamp": "2025-03-01T17:10:26"
        }
    }
}
```

Step 5 Use the **show interface brief** command to display a summary of interface statuses, including link state.

Example:

```
switch# show interface ethernet 1/1,ethernet 1/2 brief
--------------------------------------------------------------------------------
Ethernet        VLAN    Type Mode   Status  Reason                 Speed     Port
Interface                                                                    Ch #
--------------------------------------------------------------------------------
Eth1/2          --      eth  routed down    trackedPortDown          auto(D) --
Eth1/1          --      eth  routed down    Administratively down    auto(D) --
```

Link-State Tracking Configuration Examples

Link-State Tracking Configuration Examples — concept overview.

Link-State Tracking Configuration Examples

Adding and configuring link-state tracking groups and interfaces.

1. Configuring link-state tracking groups and verifying configurations

```
switch(config)# link state group track-group-1
switch(config-link-state)#
switch(config-link-state)# exit

switch# show link state group track-group-1
Link State Group: track-group-1
Status: Pending
```

```
Uplink Interfaces:
Downlink Interfaces:
Number of times disabled: 0
Last disabled: Thu Jan 01 00:00:00 1970
Downlink bringup delay timer: 60
```

2. Adding uplink interfaces to link-state tracking groups and verifying configurations

```
switch(config)# int ethernet 1/1
switch(config-if)# link state group track-group-1 type ?
  downlink  Interface associate as downlink
  uplink    Interface associate as uplink
switch(config-if)# link state group track-group-1 type uplink
switch(config-if)# exit

switch# show link state group track-group-1
Link State Group: track-group-1
Status: Pending
Uplink Interfaces: Eth1/1
Downlink Interfaces:
Number of times disabled: 0
Last disabled: Thu Jan 01 00:00:00 1970
Downlink bringup delay timer: 60
```

3. Adding downlink interfaces to link-state tracking groups and verifying configurations

```
switch(config)# interface ethernet 1/2
switch(config-if)# link state group track-group-1 type downlink
switch(config-if)# exit
```

4. Readding downlink interfaces to link state tracking group

```
switch(config-if)# interface ethernet 1/2
switch(config-if)# link state group track-group-1 type downlink

switch#show link state group track-group-1
Link State Group: track-group-1
Status: Pending
Uplink Interfaces:
Downlink Interfaces: Eth1/2
Number of times disabled: 0
Last disabled: Thu Jan 01 00:00:00 1970
Downlink bringup delay timer: 60
```

5. Readding uplink interfaces to link state tracking group

```
switch(config-if)# interface ethernet  1/1
switch(config-if)# link state group track-group-1 type uplink
switch(config-if)# exit

switch# show link state group track-group-1
Link State Group: track-group-1
Status: Up
Uplink Interfaces: Eth1/1
Downlink Interfaces: Eth1/2
Number of times disabled: 0
Last disabled: Thu Jan 01 00:00:00 1970
Downlink bringup delay timer: 60
```

Removing interfaces and resetting link-state tracking parameters.

1. Removing the custom downlink bringup delay timer and verifying the default timer

```
switch(config)# link state group track-group-1
switch(config-link-state)# no downlink bringup delay
switch(config-link-state)# exit
```

```
switch(config)# show link state group track-group-1
Link State Group: track-group-1
Status: Up
Uplink Interfaces: Eth1/1
Downlink Interfaces: Eth1/2
Number of times disabled: 1
Last disabled: Wed May 21 06:04:01 2025
Downlink bringup delay timer: 60 ------------------> default downlink bringup delay
 timer
```

2. Removing uplink interfaces from the link state tracking group

```
switch(config)# interface ethernet 1/1
switch(config-if)# no link state group track-group-1 type uplink
switch(config-if)# exit

switch# show link state group track-group-1
Link State Group: track-group-1
Status: Pending
Uplink Interfaces:
Downlink Interfaces: Eth1/2
Number of times disabled: 1
Last disabled: Wed May 21 06:04:01 2025
Downlink bringup delay timer: 60
```

3. Removing downlink interfaces from the link state tracking group

```
switch(config)# interface ethernet 1/2
switch(config-if)# no link state group track-group-1 type downlink

switch# show link state group track-group-1
Link State Group: track-group-1
Status: Pending
Uplink Interfaces:
Downlink Interfaces:
Number of times disabled: 0
Last disabled: Thu Jan 01 00:00:00 1970
Downlink bringup delay timer: 60
```

Verifying link-state tracking group status and behavior.

1. Verifying link-state tracking groups status as Up

```
switch# show link state group track-group-1
Link State Group: track-group-1
Status: Up
Uplink Interfaces: Eth1/1
Downlink Interfaces: Eth1/2
Number of times disabled: 0
Last disabled: Thu Jan 01 00:00:00 1970
Downlink bringup delay timer: 60
```

2. Verifying group status and downlink behavior after shutting down uplink interface

```
switch(config-link-state)# interface ethernet 1/1
switch(config-if)# shutdown

switch# show link state group track-group-1
Link State Group: track-group-1
Status: Down
Uplink Interfaces: Eth1/1
Downlink Interfaces: Eth1/2
Number of times disabled: 1.  ----> No. of  time disabled should increase by
```

```
 1
Last disabled: Wed May 21 06:04:01 2025        ——> last disabled timestamp
Downlink bringup delay timer: 60

switch(config-if)# show interface ethernet 1/1,  ethernet 1/2 brief
--------------------------------------------------------------------------------
Ethernet        VLAN    Type Mode   Status  Reason                 Speed     Port
Interface                                                                    Ch #
--------------------------------------------------------------------------------
Eth1/2          --      eth  routed  down    trackedPortDown          auto(D)   --
Eth1/1          --      eth  routed  down    Administratively down    auto(D)   --
```

3. Verifying group status and downlink behavior after enabling uplink interface

```
switch(config-link-state)# int ethernet 1/1
switch(config-if)# no shutdown

switch# show link state group track-group-1
Link State Group: track-group-1
Status: Up
Uplink Interfaces: Eth1/1
Downlink Interfaces: Eth1/2
Number of times disabled: 1
Last disabled: Wed May 21 06:04:01 2025
Downlink bringup delay timer: 5

switch(config)# show int ethernet 1/1,  ethernet 1/2 brief
--------------------------------------------------------------------------------
Ethernet        VLAN    Type Mode   Status  Reason                 Speed     Port
Interface                                                                    Ch #
--------------------------------------------------------------------------------
Eth1/2          --      eth  routed  up      none                   400G(D) --
Eth1/1          --      eth  routed  up      none                   400G(D) --
```

4. Configuring the downlink bringup delay timer and verifying

```
switch(config)# link state group track-group-1
switch(config-link-state)# downlink bringup delay 300
switch(config-link-state)#
switch(config-link-state)# exit

switch# show link state group track-group-1
Link State Group: track-group-1
Status: Up
Uplink Interfaces: Eth1/1
Downlink Interfaces: Eth1/2
Number of times disabled: 1
Last disabled: Wed May 21 06:04:01 2025
Downlink bringup delay timer: 300 ------>  configure downlink bringup delay
timer
```

INDEX

A

abort 147–148, 151–152, 155–156
add 78

C

clear mac address-table dynamic address 20
clear spanning-tree counters 132
clear spanning-tree counters interface 171
clear spanning-tree detected-protocol 131, 170
clear spanning-tree detected-protocol interface 170
clear vlan 55, 90
config t 12, 14, 21, 46–47, 49–51, 59–60, 75–76, 78, 80–83, 86–88, 116–117, 119–130, 146–151, 153, 155, 157–169, 184, 186, 188–191, 193–196, 198, 200–201

D

diameter 120, 155–157

F

feature private-vlan 75
feature vtp 59–60

H

hello 155
hello-time 120, 156

I

instance 151, 153–154
interface 81–84, 86–88, 123–125, 129–130, 160–162, 168–169, 186, 188, 194–195, 198, 201
interface vlan 80

M

mac address-table aging-time 18–19
mac address-table static 14
mac-address bpdu source version 2 99

N

name 49, 151–152
no private-vlan 78
no vlan 78

P

primary root 155
private-vlan mapping 90

R

remove 78
revision 150–152

S

show interface 53–54
show interface private-vlan mapping 90
show interface switchport 82–83, 85–90
show interface vlan 80–81, 90
show mac address-table 20
show mac address-table aging-time 18–19
show mac address-table static 14
show running 117–118
show running-config spanning-tree 171, 202
show running-config spanning-tree all 116, 146, 171
show running-config vlan 52, 54, 90
show spanning-tree 117–120, 129–130, 132, 169–170
show spanning-tree detail 171
show spanning-tree detail vlan 171
show spanning-tree interface 123–124, 186–189, 191–192, 198–199, 201–202
show spanning-tree mst 155–168, 171, 202
show spanning-tree mst configuration 148–154, 171
show spanning-tree mst detail 171
show spanning-tree pathcost method 124–125
show spanning-tree summary 171, 184–185, 189–190, 193, 196–197, 200, 202
show spanning-tree vlan 120–122, 126–129, 171
show system vlan reserved 41
show vlan 46–50, 54
show vlan counters 55, 90

show vlan private-vlan 76–79, 90
show vlan summary 54
show vtp counters 60–61
show vtp interface 60–61
show vtp password 60–61
show vtp status 49–50, 55, 60
show vtp trunk interface eth a/b 58
spanning-tree 123–125
spanning-tree bpdufilter disable 194
spanning-tree bpdufilter enable 194
spanning-tree bpduguard disable 190
spanning-tree bpduguard enable 190
spanning-tree guard 198
spanning-tree link-type 105, 129–130, 169
spanning-tree loopguard default 196
spanning-tree mode mst 146
spanning-tree mode rapid-pvst 116
spanning-tree mst 155, 157–162
spanning-tree mst configuration 147–151, 153
spanning-tree mst forward-time 119–120, 155, 164
spanning-tree mst hello-time 119–120, 155, 163
spanning-tree mst max-age 119–120, 155, 165–166
spanning-tree mst max-hops 166–167
spanning-tree mst pre-standard 168
spanning-tree mst priority 158
spanning-tree mst root primary 158
spanning-tree mst root secondary 158
spanning-tree mst simulate pvst 201
spanning-tree mst simulate pvst disable 201
spanning-tree pathcost method 124–125
spanning-tree port type 103
spanning-tree port type edge 185–186
spanning-tree port type edge bpdufilter default 193–194
spanning-tree port type edge bpduguard default 189–190
spanning-tree port type edge default 184
spanning-tree port type edge trunk 185–186
spanning-tree port type network 187–188
spanning-tree port type network default 184, 187
spanning-tree port type normal 186–187
spanning-tree vlan 117, 119–122, 126–129, 155
state active 49
state suspend 49
switching-mode store-forward 95
switchport 83–84, 87–88
switchport mode private-vlan host 81–82
switchport mode private-vlan promiscuous 86
switchport mode private-vlan trunk allowed vlan 87–88
switchport mode private-vlan trunk promiscuous 87–88
switchport mode private-vlan trunk secondary 83–84
switchport mode trunk 53
switchport private-vlan trunk allowed 72
switchport private-vlan trunk allowed vlan 83–84
switchport private-vlan trunk native vlan 83–84, 87–88
switchport vlan mapping 53–54
switchport vlan mapping enable 53
system vlan long-name 51–52

V

vlan 21, 41, 44, 46–49, 76–78, 151
vlan configuration 50–51
vtp domain 59–60
vtp file 59–60
vtp password 59–60
vtp version 59–60