



## **Cisco Nexus 5000 Series NX-OS Layer 2 Switching Configuration Guide, Release 5.1(3)N1(1)**

**First Published:** December 05, 2011

**Last Modified:** November 27, 2012

### **Americas Headquarters**

Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706  
USA  
<http://www.cisco.com>  
Tel: 408 526-4000  
800 553-NETS (6387)  
Fax: 408 527-0883

Text Part Number: OL-25842-01

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

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

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

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

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

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.

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

© 2011-2012 Cisco Systems, Inc. All rights reserved.



## CONTENTS

---

### Preface

#### Preface xvii

Audience xvii

Document Conventions xvii

Documentation Feedback xix

Obtaining Documentation and Submitting a Service Request xix

---

### CHAPTER 1

#### New and Changed Information 1

New and Changed Information for this Release 1

---

### CHAPTER 2

#### Overview 3

Layer 2 Ethernet Switching Overview 3

VLANs 3

Private VLANs 4

Spanning Tree 4

STP Overview 4

Rapid PVST+ 5

MST 5

STP Extensions 5

---

### CHAPTER 3

#### Configuring Ethernet Interfaces 7

Information About Ethernet Interfaces 7

Interface Command 7

Information About Unified Ports 8

Guidelines and Limitations for Unified Ports 8

Unidirectional Link Detection Parameter 8

Default UDLD Configuration 9

UDLD Aggressive and Nonaggressive Modes 10

Interface Speed	10
Cisco Discovery Protocol	10
Default CDP Configuration	10
Error-Disabled State	11
About Port Profiles	11
Guidelines and Limitations for Port Profiles	12
Debounce Timer Parameters	13
MTU Configuration	13
Configuring Ethernet Interfaces	13
Configuring a Layer 3 Interface on a Cisco Nexus Device	13
Configuring Unified Ports	14
Configuring the UDLD Mode	15
Configuring Interface Speed	17
Disabling Link Negotiation	17
Configuring the CDP Characteristics	18
Enabling or Disabling CDP	19
Enabling the Error-Disabled Detection	20
Enabling the Error-Disabled Recovery	21
Configuring the Error-Disabled Recovery Interval	22
Port Profiles	23
Creating a Port Profile	23
Modifying a Port Profile	24
Enabling a Specific Port Profile	26
Inheriting a Port Profile	27
Removing an Inherited Port Profile	28
Assigning a Port Profile to a Range of Interfaces	30
Removing a Port Profile from a Range of Interfaces	30
Configuration Examples for Port Profiles	31
Configuring the Debounce Timer	32
Configuring the Description Parameter	33
Disabling and Restarting Ethernet Interfaces	34
Displaying Interface Information	34
Default Physical Ethernet Settings	37

---

**CHAPTER 4****Configuring VLANs 39**

Information About VLANs	39
Understanding VLANs	39
Understanding VLAN Ranges	40
Creating, Deleting, and Modifying VLANs	42
About the VLAN Trunking Protocol	42
Guidelines and Limitations for VTP	43
Configuring a VLAN	43
Creating and Deleting a VLAN	43
Changing the Range of Reserved VLANs	44
Configuring a VLAN	46
Adding Ports to a VLAN	47
Configuring VTP	48
Verifying the VLAN Configuration	49
<hr/>	
<b>CHAPTER 5</b>	<b>Configuring Private VLANs 51</b>
Information About Private VLANs	51
Primary and Secondary VLANs in Private VLANs	52
Private VLAN Ports	52
Primary, Isolated, and Community Private VLANs	53
Associating Primary and Secondary VLANs	54
Private VLAN Promiscuous Trunks	55
Private VLAN Isolated Trunks	55
Broadcast Traffic in Private VLANs	56
Private VLAN Port Isolation	56
Guidelines and Limitations for Private VLANs	56
Configuring a Private VLAN	57
Enabling Private VLANs	57
Configuring a VLAN as a Private VLAN	57
Associating Secondary VLANs with a Primary Private VLAN	58
Configuring an Interface as a Private VLAN Host Port	60
Configuring an Interface as a Private VLAN Promiscuous Port	61
Configuring a Promiscuous Trunk Port	62
Configuring an Isolated Trunk Port	63
Configuring Private VLANs on FEX Trunk Ports	64
Configuring the Allowed VLANs for PVLAN Trunking Ports	65

Configuring Native 802.1Q VLANs on Private VLANs	66
Verifying the Private VLAN Configuration	67

---

**CHAPTER 6****Configuring Access and Trunk Interfaces 69**

Information About Access and Trunk Interfaces	69
Understanding Access and Trunk Interfaces	69
Understanding IEEE 802.1Q Encapsulation	70
Understanding Access VLANs	71
Understanding the Native VLAN ID for Trunk Ports	72
Understanding Allowed VLANs	72
Understanding Native 802.1Q VLANs	72
Configuring Access and Trunk Interfaces	73
Configuring a LAN Interface as an Ethernet Access Port	73
Configuring Access Host Ports	74
Configuring Trunk Ports	74
Configuring the Native VLAN for 802.1Q Trunking Ports	75
Configuring the Allowed VLANs for Trunking Ports	76
Configuring Native 802.1Q VLANs	77
Verifying the Interface Configuration	78

---

**CHAPTER 7****Configuring Port Channels 79**

Information About Port Channels	79
Understanding Port Channels	79
Guidelines and Limitations for Port Channel Configuration	80
Compatibility Requirements	81
Load Balancing Using Port Channels	82
Understanding LACP	85
LACP Overview	85
LACP ID Parameters	86
Channel Modes	86
LACP Marker Responders	87
LACP-Enabled and Static Port Channel Differences	88
Configuring Port Channels	88
Creating a Port Channel	88
Adding a Port to a Port Channel	89

Configuring Load Balancing Using Port Channels	90
Configuring Hardware Hashing for Multicast Traffic	91
Enabling LACP	91
Configuring the Channel Mode for a Port	92
Configuring the LACP Fast Timer Rate	93
Configuring the LACP System Priority and System ID	94
Configuring the LACP Port Priority	95
Disabling LACP Graceful Convergence	96
Reenabling LACP Graceful Convergence	97
Verifying Port Channel Configuration	98
Verifying the Load-Balancing Outgoing Port ID	99

**CHAPTER 8****Configuring Virtual Port Channels 101**

Information About vPCs	101
vPC Overview	101
Terminology	103
vPC Terminology	103
Fabric Extender Terminology	103
Supported vPC Topologies	104
Cisco Nexus Device vPC Topology	104
Single Homed Fabric Extender vPC Topology	104
Dual Homed Fabric Extender vPC Topology	105
vPC Domain	105
Peer-Keepalive Link and Messages	106
Compatibility Parameters for vPC Peer Links	106
Configuration Parameters That Must Be Identical	107
Configuration Parameters That Should Be Identical	108
Graceful Type-1 Check	109
Per-VLAN Consistency Check	109
vPC Auto-Recovery	109
vPC Peer Links	109
vPC Peer Link Overview	109
vPC Number	110
vPC Interactions with Other Features	111
Configuring vPC Peer Links and Links to the Core	111

vPC and LACP	113
vPC Peer Links and STP	113
vPC and ARP	114
CFSoE	114
vPC Peer Switch	115
Guidelines and Limitations for vPCs	115
Configuring vPCs	116
Enabling vPCs	116
Disabling vPCs	117
Creating a vPC Domain	118
Configuring a vPC Keepalive Link and Messages	118
Creating a vPC Peer Link	121
Checking the Configuration Compatibility	122
Enabling vPC Auto-Recovery	123
Configuring the Restore Time Delay	124
Excluding VLAN Interfaces from Shutting Down a vPC Peer Link Fails	125
Configuring the VRF Name	126
Binding a VRF Instance to a vPC	126
Enabling Layer 3 Forwarding to the Gateway MAC Address of the vPC	127
Suspending Orphan Ports on a Secondary Switch in a vPC Topology	128
Creating an EtherChannel Host Interface	129
Moving Other Port Channels into a vPC	130
Manually Configuring a vPC Domain MAC Address	131
Manually Configuring the System Priority	132
Manually Configuring a vPC Peer Switch Role	133
Configuring the vPC Peer Switch	134
Configuring a Pure vPC Peer Switch Topology	134
Configuring a Hybrid vPC Peer Switch Topology	136
Verifying the vPC Configuration	138
Viewing the Graceful Type-1 Check Status	138
Viewing a Global Type-1 Inconsistency	139
Viewing an Interface-Specific Type-1 Inconsistency	140
Viewing a Per-VLAN Consistency Status	141
vPC Example Configurations	143
Dual Homed Fabric Extender vPC Configuration Example	143

Single Homed Fabric Extender vPC Configuration Example	145
vPC Default Settings	147

---

**CHAPTER 9****Configuring Enhanced Virtual Port Channels 149**

Information About Enhanced vPCs	150
Enhanced Virtual Port Channels Overview	150
Supported Platforms and Topologies	150
Enhanced vPC Scalability	151
Enhanced vPC Failure Response	151
Licensing Requirements for Enhanced vPC	152
Configuring Enhanced vPCs	152
Overview of Configuration Steps for Enhanced vPC	152
Verifying Enhanced vPCs	153
Verifying the Enhanced vPC Configuration	153
Verifying the Consistency of Port Channel Numbers	154
Verifying Common Port Channel Members	155
Verifying Interface Level Consistency for Enhanced vPCs	156
Enhanced vPC Example Configuration	157

---

**CHAPTER 10****Configuring Rapid PVST+ 161**

Information About Rapid PVST+	161
Understanding STP	161
STP Overview	161
Understanding How a Topology is Created	162
Understanding the Bridge ID	162
Bridge Priority Value	162
Extended System ID	163
STP MAC Address Allocation	163
Understanding BPDUs	164
Election of the Root Bridge	165
Creating the Spanning Tree Topology	165
Understanding Rapid PVST+	166
Rapid PVST+ Overview	166
Rapid PVST+ BPDUs	167
Proposal and Agreement Handshake	168

Protocol Timers	169
Port Roles	169
Port States	170
Rapid PVST+ Port State Overview	170
Blocking State	171
Learning State	171
Forwarding State	171
Disabled State	172
Summary of Port States	172
Synchronization of Port Roles	172
Processing Superior BPDU Information	173
Processing Inferior BPDU Information	174
Spanning-Tree Dispute Mechanism	174
Port Cost	174
Port Priority	175
Rapid PVST+ and IEEE 802.1Q Trunks	175
Rapid PVST+ Interoperation with Legacy 802.1D STP	175
Rapid PVST+ Interoperation with 802.1s MST	176
Configuring Rapid PVST+	176
Enabling Rapid PVST+	176
Enabling Rapid PVST+ per VLAN	177
Configuring the Root Bridge ID	178
Configuring a Secondary Root Bridge	179
Configuring the Rapid PVST+ Port Priority	180
Configuring the Rapid PVST+ Path-Cost Method and Port Cost	181
Configuring the Rapid PVST+ Bridge Priority of a VLAN	182
Configuring the Rapid PVST+ Hello Time for a VLAN	183
Configuring the Rapid PVST+ Forward Delay Time for a VLAN	183
Configuring the Rapid PVST+ Maximum Age Time for a VLAN	184
Specifying the Link Type	185
Restarting the Protocol	185
Verifying the Rapid PVST+ Configuration	186
<hr/>	
<b>CHAPTER 11</b>	<b>Configuring Multiple Spanning Tree 187</b>
	Information About MST 187

MST Overview	187
MST Regions	188
MST BPDUs	188
MST Configuration Information	189
IST, CIST, and CST	189
IST, CIST, and CST Overview	189
Spanning Tree Operation Within an MST Region	190
Spanning Tree Operations Between MST Regions	190
MST Terminology	191
Hop Count	192
Boundary Ports	192
Spanning-Tree Dispute Mechanism	193
Port Cost and Port Priority	194
Interoperability with IEEE 802.1D	194
Interoperability with Rapid PVST+: Understanding PVST Simulation	195
Configuring MST	195
MST Configuration Guidelines	195
Enabling MST	195
Entering MST Configuration Mode	196
Specifying the MST Name	197
Specifying the MST Configuration Revision Number	198
Specifying the Configuration on an MST Region	199
Mapping and Unmapping VLANs to MST Instances	200
Mapping Secondary VLANs to Same MSTI as Primary VLANs for Private VLANs	201
Configuring the Root Bridge	202
Configuring a Secondary Root Bridge	203
Configuring the Port Priority	204
Configuring the Port Cost	205
Configuring the Switch Priority	206
Configuring the Hello Time	207
Configuring the Forwarding-Delay Time	208
Configuring the Maximum-Aging Time	208
Configuring the Maximum-Hop Count	209
Configuring PVST Simulation Globally	210
Configuring PVST Simulation Per Port	210

Specifying the Link Type	211
Restarting the Protocol	212
Verifying the MST Configuration	213

**CHAPTER 12****Configuring STP Extensions 215**

Overview	215
Information About STP Extensions	215
Understanding STP Port Types	215
Spanning Tree Edge Ports	215
Spanning Tree Network Ports	216
Spanning Tree Normal Ports	216
Understanding Bridge Assurance	216
Understanding BPDU Guard	216
Understanding BPDU Filtering	217
Understanding Loop Guard	218
Understanding Root Guard	219
Configuring STP Extensions	219
STP Extensions Configuration Guidelines	219
Configuring Spanning Tree Port Types Globally	219
Configuring Spanning Tree Edge Ports on Specified Interfaces	220
Configuring Spanning Tree Network Ports on Specified Interfaces	222
Enabling BPDU Guard Globally	223
Enabling BPDU Guard on Specified Interfaces	223
Enabling BPDU Filtering Globally	224
Enabling BPDU Filtering on Specified Interfaces	225
Enabling Loop Guard Globally	227
Enabling Loop Guard or Root Guard on Specified Interfaces	227
Verifying the STP Extension Configuration	228

**CHAPTER 13****Configuring Flex Links 229**

Information About Flex Links	229
Guidelines and Limitations for Flex Link	230
Default Settings for Flex Link	231
Configuring Flex Links	232
Configuring Flex Link Preemption	233

Verifying Flex Link Configuration 235

Flex Link Configuration Examples 236

---

**CHAPTER 14****Configuring LLDP 239**

Configuring LLDP 239

Configuring Interface LLDP 240

---

**CHAPTER 15****Configuring MAC Address Tables 243**

Information About MAC Addresses 243

Configuring MAC Addresses 243

Configuring Static MAC Addresses 243

Configuring the Aging Time for the MAC Table 244

Clearing Dynamic Addresses from the MAC Table 245

Verifying the MAC Address Configuration 245

---

**CHAPTER 16****Configuring IGMP Snooping 247**

Information About IGMP Snooping 247

IGMPv1 and IGMPv2 248

IGMPv3 249

IGMP Snooping Querier 249

IGMP Forwarding 249

Configuring IGMP Snooping Parameters 250

Verifying the IGMP Snooping Configuration 253

---

**CHAPTER 17****Configuring MVR 255**

Information About MVR 255

MVR Overview 255

MVR Interoperation with Other Features 256

Licensing Requirements for MVR 256

Guidelines and Limitations for MVR 256

Default MVR Settings 257

Configuring MVR 257

Configuring MVR Global Parameters 257

Configuring MVR Interfaces 259

Verifying the MVR Configuration 260

---

**CHAPTER 18****Configuring Traffic Storm Control 263**

- Information About Traffic Storm Control 263
- Guidelines and Limitations for Traffic Storm Control 265
- Configuring Traffic Storm Control 266
  - Verifying the Traffic Storm Control Configuration 266
- Traffic Storm Control Example Configuration 267
- Default Settings for Traffic Storm Control 267

---

**CHAPTER 19****Configuring the Fabric Extender 269**

- Information About the Cisco Nexus 2000 Series Fabric Extender 269
- Fabric Extender Terminology 270
- Fabric Extender Features 271
  - Layer 2 Host Interfaces 271
  - Host Port Channel 271
  - VLANs and Private VLANs 272
  - Virtual Port Channels 272
  - Fibre Channel over Ethernet Support 273
  - Protocol Offload 273
  - Quality of Service 274
  - Access Control Lists 274
  - IGMP Snooping 274
  - Switched Port Analyzer 274
  - Fabric Interface Features 274
- Oversubscription 274
- Management Model 274
- Forwarding Model 275
- Connection Model 275
  - Static Pinning Fabric Interface Connection 276
  - Port Channel Fabric Interface Connection 277
- Port Numbering Convention 278
- Fabric Extender Image Management 278
- Fabric Extender Hardware 278
  - Chassis 278
  - Ethernet Interfaces 278

Associating a Fabric Extender to a Fabric Interface	279
Associating a Fabric Extender to an Ethernet Interface	279
Associating a Fabric Extender to a Port Channel	280
Disassociating a Fabric Extender from an Interface	282
Configuring Fabric Extender Global Features	283
Enabling the Fabric Extender Locator LED	285
Redistributing the Links	285
Changing the Number of Links	286
Maintaining the Pinning Order	286
Redistributing Host Interfaces	286
Verifying the Fabric Extender Configuration	287
Verifying the Chassis Management Information	290
Configuring the Cisco Nexus N2248TP-E Fabric Extender	294
Configuring the Shared Buffer	295
Configuring the Queue-Limit at the Global Level	296
Configuring the Queue-Limit at the Port Level	297
Configuring Uplink Distance	298





## Preface

---

The Preface contains the following sections:

- [Audience](#), page xvii
- [Document Conventions](#), page xvii
- [Documentation Feedback](#), page xix
- [Obtaining Documentation and Submitting a Service Request](#), page xix

## Audience

This publication is for network administrators who configure and maintain Cisco Nexus devices and Cisco Nexus 2000 Series Fabric Extenders.

## Document Conventions



### Note

As part of our constant endeavor to remodel our documents to meet our customers' requirements, we have modified the manner in which we document configuration tasks. As a result of this, you may find a deviation in the style used to describe these tasks, with the newly included sections of the document following the new format.

Command descriptions use the following conventions:

Convention	Description
<b>bold</b>	Bold text indicates the commands and keywords that you enter literally as shown.
<i>Italic</i>	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element (keyword or argument).

Convention	Description
[x   y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x   y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.
[x {y   z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.
<i>variable</i>	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Examples use the following conventions:

Convention	Description
<code>screen font</code>	Terminal sessions and information the switch displays are in screen font.
<b><code>boldface screen font</code></b>	Information you must enter is in boldface screen font.
<i><code>italic screen font</code></i>	Arguments for which you supply values are in italic screen font.
<>	Nonprinting characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

This document uses the following conventions:



**Note**

Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.



**Caution**

Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

## Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to: [ciscodfa-docfeedback@cisco.com](mailto:ciscodfa-docfeedback@cisco.com).

We appreciate your feedback.

## Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, using the Cisco Bug Search Tool (BST), submitting a service request, and gathering additional information, see *What's New in Cisco Product Documentation*, at: <http://www.cisco.com/c/en/us/td/docs/general/whatsnew/whatsnew.html>.

To receive new and revised Cisco technical content directly to your desktop, you can subscribe to the [What's New in Cisco Product Documentation RSS feed](#). RSS feeds are a free service.





# CHAPTER 1

## New and Changed Information

---

This chapter contains the following sections:

- [New and Changed Information for this Release](#) , page 1

## New and Changed Information for this Release

The following table provides an overview of the significant changes to this guide for this current release. The table does not provide an exhaustive list of all changes made to the configuration guides or of the new features in this release.

**Table 1: New Features**

Feature	Description	Where Documented
PVLAN Trunk Over FEX Ports	Enables PVLAN trunk over FEX ports	<a href="#">Configuring Private VLANs</a> , on page 51





## Overview

---

This chapter contains the following sections:

- [Layer 2 Ethernet Switching Overview, page 3](#)
- [VLANs, page 3](#)
- [Private VLANs, page 4](#)
- [Spanning Tree, page 4](#)

## 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 assigns a domain (for example, a server) to each device to solve traffic congestion caused by high-bandwidth devices and large number of users.

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. 1/10-Gigabit Ethernet operates in full-duplex only.

## 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 comes up.

The devices support 4094 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.

---

## Private VLANs

Private VLANs provide traffic separation and security at the Layer 2 level.

A private VLAN is one or more pairs of a primary VLAN and a secondary VLAN, all with the same primary VLAN. The two types of secondary VLANs are isolated and community VLANs. Hosts on isolated VLANs communicate only with hosts in the primary VLAN. Hosts in a community VLAN can communicate only among themselves and with hosts in the primary VLAN but not with hosts in isolated VLANs or in other community VLANs.

Regardless of the combination of isolated and community secondary VLANs, all interfaces within the primary VLAN comprise one Layer 2 domain, and therefore, require only one IP subnet.

## Spanning Tree

This section discusses the implementation of the Spanning Tree Protocol (STP). 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).

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 device runs Rapid PVST+ and MST. You can use either Rapid PVST+ or MST in a given VDC; you cannot mix both in one VDC. Rapid PVST+ is the default STP protocol.



---

**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 prevents the occurrence of loop bridging because of unidirectional link failure in a point-to-point link.
- Root Guard—Root guard prevents a port from becoming a root port or a blocked port. If you configure a port with root guard then the port receives a superior BPDU and it immediately goes to root-inconsistent (blocked) state.





## Configuring Ethernet Interfaces

---

This chapter contains the following sections:

- [Information About Ethernet Interfaces, page 7](#)
- [Configuring Ethernet Interfaces, page 13](#)
- [Displaying Interface Information, page 34](#)
- [Default Physical Ethernet Settings, page 37](#)

### Information About Ethernet Interfaces

The Ethernet ports can operate as standard Ethernet interfaces connected to servers or to a LAN.

The Ethernet interfaces also support Fibre Channel over Ethernet (FCoE). FCoE allows the physical Ethernet link to carry both Ethernet and Fibre Channel traffic.

The Ethernet interfaces are enabled by default.

### Interface Command

You can enable the various capabilities of the Ethernet interfaces on a per-interface basis using the **interface** command. When you enter the **interface** command, you specify the following information:

- Interface type—All physical Ethernet interfaces use the **ethernet** keyword.
- Slot number:
  - Slot 1 includes all the fixed ports.
  - Slot 2 includes the ports on the upper expansion module (if populated).
  - Slot 3 includes the ports on the lower expansion module (if populated).
  - Slot 4 includes the ports on the lower expansion module (if populated).
- Port number— Port number within the group.

The interface numbering convention is extended to support use with a Cisco Nexus Fabric Extender as follows:

```
switch(config)# interface ethernet [chassis]/slot/port
```

- The chassis ID is an optional entry that you can use to address the ports of a connected Fabric Extender. The chassis ID is configured on a physical Ethernet or EtherChannel interface on the switch to identify the Fabric Extender discovered through the interface. The chassis ID ranges from 100 to 199.

## Information About Unified Ports

Cisco Nexus unified ports allow you to configure a physical port on a Cisco Nexus device switch as a 1/10-Gigabit Ethernet, Fibre Channel over Ethernet (FCoE), or 2-, 4-, 8-Gigabit native Fibre Channel port.

Currently, most networks have two types of switches for different types of networks. For example, LAN switches carry Ethernet traffic up to Catalyst or Nexus switches carry FC traffic from servers to MDS switches. With unified port technology, you can deploy a unified platform, unified device, and unified wire approach. Unified ports allow you to move from an existing segregated platform approach where you choose LAN and SAN port options to transition to a single, unified fabric that is transparent and consistent with existing practices and management software. A unified fabric includes the following:

- Unified platform—Uses the same hardware platform and the same software code level and certifies it once for your LAN and SAN environments.
- Unified device—Runs LAN and SAN services on the same platform switch. The unified device allows you to connect your Ethernet and Fibre Channel cables to the same device.
- Unified wire—Converges LAN and SAN networks on a single converged network adapter (CNA) and connects them to your server.

A unified fabric allows you to manage Ethernet and FCoE features independently with existing Cisco tools.

## Guidelines and Limitations for Unified Ports

- Ethernet ports and Fibre Channel ports must be configured in the following order:
  - Fibre Channel ports must be configured from the last port of the module.
  - Ethernet ports must be configured from the first port of the module.

If the order is not followed, the following errors are displayed:

```
ERROR: Ethernet range starts from first port of the module
ERROR: FC range should end on last port of the module
```

## Unidirectional Link Detection Parameter

The Cisco-proprietary Unidirectional Link Detection (UDLD) protocol allows ports that are connected through fiber optics or copper (for example, Category 5 cabling) Ethernet cables to monitor the physical configuration of the cables and detect when a unidirectional link exists. When the switch detects a unidirectional link, UDLD shuts down the affected LAN port and alerts the user. Unidirectional links can cause a variety of problems, including spanning tree topology loops.

UDLD is a Layer 2 protocol that works with the Layer 1 protocols to determine the physical status of a link. At Layer 1, autonegotiation takes care of physical signaling and fault detection. UDLD performs tasks that

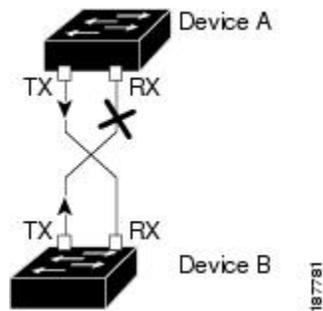
autonegotiation cannot perform, such as detecting the identities of neighbors and shutting down misconnected LAN ports. When you enable both autonegotiation and UDLD, Layer 1 and Layer 2 detections work together to prevent physical and logical unidirectional connections and the malfunctioning of other protocols.

A unidirectional link occurs whenever traffic transmitted by the local device over a link is received by the neighbor but traffic transmitted from the neighbor is not received by the local device. If one of the fiber strands in a pair is disconnected, and if autonegotiation is active, the link does not stay up. In this case, the logical link is undetermined, and UDLD does not take any action. If both fibers are working normally at Layer 1, then UDLD at Layer 2 determines whether those fibers are connected correctly and whether traffic is flowing bidirectionally between the correct neighbors. This check cannot be performed by autonegotiation, because autonegotiation operates at Layer 1.

A Cisco Nexus device periodically transmits UDLD frames to neighbor devices on LAN ports with UDLD enabled. If the frames are echoed back within a specific time frame and they lack a specific acknowledgment (echo), the link is flagged as unidirectional and the LAN port is shut down. Devices on both ends of the link must support UDLD in order for the protocol to successfully identify and disable unidirectional links.

The following figure shows an example of a unidirectional link condition. Device B successfully receives traffic from Device A on the port. However, Device A does not receive traffic from Device B on the same port. UDLD detects the problem and disables the port.

**Figure 1: Unidirectional Link**



### Default UDLD Configuration

The following table shows the default UDLD configuration.

**Table 2: UDLD Default Configuration**

Feature	Default Value
UDLD global enable state	Globally disabled
UDLD aggressive mode	Disabled
UDLD per-port enable state for fiber-optic media	Enabled on all Ethernet fiber-optic LAN ports
UDLD per-port enable state for twisted-pair (copper) media	Enabled

## UDLD Aggressive and Nonaggressive Modes

UDLD aggressive mode is disabled by default. You can configure UDLD aggressive mode only on point-to-point links between network devices that support UDLD aggressive mode. If UDLD aggressive mode is enabled, when a port on a bidirectional link that has a UDLD neighbor relationship established stops receiving UDLD frames, UDLD tries to reestablish the connection with the neighbor. After eight failed retries, the port is disabled.

To prevent spanning tree loops, nonaggressive UDLD with the default interval of 15 seconds is fast enough to shut down a unidirectional link before a blocking port transitions to the forwarding state (with default spanning tree parameters).

When you enable the UDLD aggressive mode, the following occurs:

- One side of a link has a port stuck (both transmission and receive)
- One side of a link remains up while the other side of the link is down

In these cases, the UDLD aggressive mode disables one of the ports on the link, which prevents traffic from being discarded.

## Interface Speed

### Cisco Discovery Protocol

The Cisco Discovery Protocol (CDP) is a device discovery protocol that runs over Layer 2 (the data link layer) on all Cisco-manufactured devices (routers, bridges, access servers, and switches) and allows network management applications to discover Cisco devices that are neighbors of already known devices. With CDP, network management applications can learn the device type and the Simple Network Management Protocol (SNMP) agent address of neighboring devices that are running lower-layer, transparent protocols. This feature enables applications to send SNMP queries to neighboring devices.

CDP runs on all media that support Subnetwork Access Protocol (SNAP). Because CDP runs over the data-link layer only, two systems that support different network-layer protocols can learn about each other.

Each CDP-configured device sends periodic messages to a multicast address, advertising at least one address at which it can receive SNMP messages. The advertisements also contain time-to-live, or holdtime information, which is the length of time a receiving device holds CDP information before discarding it. Each device also listens to the messages sent by other devices to learn about neighboring devices.

The switch supports both CDP Version 1 and Version 2.

### Default CDP Configuration

The following table shows the default CDP configuration.

**Table 3: Default CDP Configuration**

Feature	Default Setting
CDP interface state	Enabled

Feature	Default Setting
CDP timer (packet update frequency)	60 seconds
CDP holdtime (before discarding)	180 seconds
CDP Version-2 advertisements	Enabled

## Error-Disabled State

An interface is in the error-disabled (err-disabled) state when the interface is enabled administratively (using the **no shutdown** command) but disabled at runtime by any process. For example, if UDLD detects a unidirectional link, the interface is shut down at runtime. However, because the interface is administratively enabled, the interface status displays as err-disabled. Once an interface goes into the err-disabled state, you must manually reenabling it or you can configure an automatic timeout recovery value. The err-disabled detection is enabled by default for all causes. The automatic recovery is not configured by default.

When an interface is in the err-disabled state, use the **errdisable detect cause** command to find information about the error.

You can configure the automatic err-disabled recovery timeout for a particular err-disabled cause by changing the time variable.

The **errdisable recovery cause** command provides automatic recovery after 300 seconds. To change the recovery period, use the **errdisable recovery interval** command to specify the timeout period. You can specify 30 to 65535 seconds.

If you do not enable the err-disabled recovery for the cause, the interface stays in the err-disabled state until you enter the **shutdown** and **no shutdown** commands. If the recovery is enabled for a cause, the interface is brought out of the err-disabled state and allowed to retry operation once all the causes have timed out. Use the **show interface status err-disabled** command to display the reason behind the error.

## About Port Profiles

You can create a port profile that contains many interface commands and apply that port profile to a range of interfaces on the Cisco Nexus device. Port profiles can be applied to the following interface types:

- Ethernet
- VLAN network interface
- Port channel

A command that is included in a port profile can be configured outside of the port profile. If the new configuration in the port profile conflicts with the configurations that exist outside the port profile, the commands configured for an interface in configuration terminal mode have higher priority than the commands in the port profile. If changes are made to the interface configuration after a port profile is attached to it, and the configuration conflicts with that in the port profile, the configurations in the interface will be given priority.

You inherit the port profile when you attach the port profile to an interface or range of interfaces. When you attach, or inherit, a port profile to an interface or range of interfaces, the switch applies all the commands in that port profile to the interfaces.

You can have one port profile inherit the settings from another port profile. Inheriting another port profile allows the initial port profile to assume all of the commands of the second, inherited, port profile that do not conflict with the initial port profile. Four levels of inheritance are supported. The same port profile can be inherited by any number of port profiles.

To apply the port profile configurations to the interfaces, you must enable the specific port profile. You can configure and inherit a port profile onto a range of interfaces prior to enabling the port profile; you then enable that port profile for the configurations to take effect on the specified interfaces.

When you remove a port profile from a range of interfaces, the switch undoes the configuration from the interfaces first and then removes the port profile link itself. When you remove a port profile, the switch checks the interface configuration and either skips the port profile commands that have been overridden by directly entered interface commands or returns the command to the default value.

If you want to delete a port profile that has been inherited by other port profiles, you must remove the inheritance before you can delete the port profile.

You can choose a subset of interfaces from which to remove a port profile from among that group of interfaces that you originally applied the profile. For example, if you configured a port profile and configured ten interfaces to inherit that port profile, you can remove the port profile from just some of the specified ten interfaces. The port profile continues to operate on the remaining interfaces to which it is applied.

If you delete a specific configuration for a specified range of interfaces using the interface configuration mode, that configuration is also deleted from the port profile for that range of interfaces only. For example, if you have a channel group inside a port profile and you are in the interface configuration mode and you delete that port channel, the specified port channel is also deleted from the port profile as well.

After you inherit a port profile on an interface or range of interfaces and you delete a specific configuration value, that port profile configuration will not operate on the specified interfaces.

If you attempt to apply a port profile to the wrong type of interface, the switch returns an error.

When you attempt to enable, inherit, or modify a port profile, the switch creates a checkpoint. If the port profile configuration fails, the switch rolls back to the prior configuration and returns an error. A port profile is never only partially applied.

## Guidelines and Limitations for Port Profiles

Port profiles have the following configuration guidelines and limitations:

- Each port profile must have a unique name across interface types and the network.
- Commands that you enter under the interface mode take precedence over the port profile's commands if there is a conflict. However, the port profile retains that command in the port profile.
- The port profile's commands take precedence over the default commands on the interface, unless the default command explicitly overrides the port profile command.
- After you inherit a port profile onto an interface or range of interfaces, you can override individual configuration values by entering the new value at the interface configuration level. If you remove the individual configuration values at the interface configuration level, the interface uses the values in the port profile again.
- There are no default configurations associated with a port profile.
- A subset of commands are available under the port profile configuration mode, depending on which interface type that you specify.

- You cannot use port profiles with Session Manager.

## Debounce Timer Parameters

The port debounce time is the amount of time that an interface waits to notify the supervisor of a link going down. During this time, the interface waits to see if the link comes back up. The wait period is a time when traffic is stopped.

You can enable the debounce timer for each interface and specify the delay time in milliseconds.



### Caution

When you enable the port debounce timer the link up and link down detections are delayed, resulting in a loss of traffic during the debounce period. This situation might affect the convergence and reconvergence of some protocols.

## MTU Configuration

The Cisco Nexus device switch does not fragment frames. As a result, the switch cannot have two ports in the same Layer 2 domain with different maximum transmission units (MTUs). A per-physical Ethernet interface MTU is not supported. Instead, the MTU is set according to the QoS classes. You modify the MTU by setting class and policy maps.



### Note

When you show the interface settings, a default MTU of 1500 is displayed for physical Ethernet interfaces and a receive data field size of 2112 is displayed for Fibre Channel interfaces.

## Configuring Ethernet Interfaces

The section includes the following topics:

### Configuring a Layer 3 Interface on a Cisco Nexus Device

On Cisco Nexus devices, you can configure a Layer 3 interface.

You can change a Layer 3 interface into a Layer 2 interface by using the **switchport** command. You can change a Layer 2 interface into a Layer 3 interface by using the **no switchport** command.

#### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface ethernet slot/port**
3. switch(config-if)# **no switchport**
4. switch(config-if)# **no shutdown**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters configuration mode.
<b>Step 2</b>	switch(config)# <b>interface ethernet slot/port</b>	Enters configuration mode for the specified interface. <b>Note</b> If this is a QSFP+ GEM, the <i>slot/port</i> syntax is <i>slot/QSFP-module/port</i> .
<b>Step 3</b>	switch(config-if)# <b>no switchport</b>	Selects the Layer 3 interface.
<b>Step 4</b>	switch(config-if)# <b>no shutdown</b>	Restarts the interface.

This example shows how to configure a Layer 3 interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)# no shutdown
```

## Configuring Unified Ports

### Before You Begin

Confirm that you have a supported Cisco Nexus switch. Unified Ports are available on the following Cisco Nexus switches:

If you're configuring a unified port as Fibre Channel or FCoE, confirm that you have enabled the **feature fcoe** command.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config) # **slot slot number**
3. switch(config-slot) # **port port number type {ethernet | fc}**
4. switch(config-slot) # **copy running-config startup-config**
5. switch(config-slot) # **reload**
6. switch(config) # **slot slot number**
7. switch(config-slot) # **no port port number type fc**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.

	Command or Action	Purpose
<b>Step 2</b>	switch(config) # <b>slot</b> <i>slot number</i>	Identifies the slot on the switch.
<b>Step 3</b>	switch(config-slot) # <b>port</b> <i>port number</i> <b>type</b> { <b>ethernet</b>   <b>fc</b> }	Configures a unified port as a native Fibre Channel port and an Ethernet port. <ul style="list-style-type: none"> <li>• <b>type</b>—Specifies the type of port to configure on a slot in a chassis.</li> <li>• <b>ethernet</b>—Specifies an Ethernet port.</li> <li>• <b>fc</b>—Specifies a Fibre Channel (FC) port.</li> </ul> <p><b>Note</b> Changing unified ports on an expansion module (GEM) requires that you power cycle the GEM card. You do not have to reboot the entire switch for changes to take effect.</p>
<b>Step 4</b>	switch(config-slot) # <b>copy running-config startup-config</b>	Copies the running configuration to the startup configuration.
<b>Step 5</b>	switch(config-slot) # <b>reload</b>	Reboots the switch.
<b>Step 6</b>	switch(config) # <b>slot</b> <i>slot number</i>	Identifies the slot on the switch.
<b>Step 7</b>	switch(config-slot) # <b>no port</b> <i>port number</i> <b>type</b> <b>fc</b>	Removes the unified port.

This example shows how to configure 20 ports as Ethernet ports and 12 as FC ports:

```
switch# configure terminal
switch(config)# slot 1
switch(config-slot)# port 21-32 type fc
switch(config-slot)# copy running-config startup-config
switch(config-slot)# reload
```

## Configuring the UDLD Mode

You can configure normal or aggressive unidirectional link detection (UDLD) modes for Ethernet interfaces on devices configured to run UDLD. Before you can enable a UDLD mode for an interface, you must make sure that UDLD is already enabled on the device that includes the interface. UDLD must also be enabled on the other linked interface and its device.

To use the normal UDLD mode, you must configure one of the ports for normal mode and configure the other port for the normal or aggressive mode. To use the aggressive UDLD mode, you must configure both ports for the aggressive mode.



### Note

Before you begin, UDLD must be enabled for the other linked port and its device.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **feature udld**
3. switch(config)# **no feature udld**
4. switch(config)# **show udld global**
5. switch(config)# **interface type slot/port**
6. switch(config-if)# **udld {enable | disable | aggressive}**
7. switch(config-if)# **show udld interface**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>feature udld</b>	Enables UDLD for the device.
<b>Step 3</b>	switch(config)# <b>no feature udld</b>	Disables UDLD for the device.
<b>Step 4</b>	switch(config)# <b>show udld global</b>	Displays the UDLD status for the device.
<b>Step 5</b>	switch(config)# <b>interface type slot/port</b>	Specifies an interface to configure, and enters interface configuration mode.
<b>Step 6</b>	switch(config-if)# <b>udld {enable   disable   aggressive}</b>	Enables the normal UDLD mode, disables UDLD, or enables the aggressive UDLD mode.
<b>Step 7</b>	switch(config-if)# <b>show udld interface</b>	Displays the UDLD status for the interface.

This example shows how to enable UDLD for the switch:

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

This example shows how to enable the normal UDLD mode for an Ethernet port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# udld enable
```

This example shows how to enable the aggressive UDLD mode for an Ethernet port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# udld aggressive
```

This example shows how to disable UDLD for an Ethernet port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
```

```
switch(config-if)# udld disable
```

This example shows how to disable UDLD for the switch:

```
switch# configure terminal
switch(config)# no feature udld
```

## Configuring Interface Speed



### Note

If the interface and transceiver speed is mismatched, the SFP validation failed message is displayed when you enter the **show interface ethernet slot/port** command. For example, if you insert a 1-Gigabit SFP transceiver into a port without configuring the speed 1000 command, you will get this error. By default, all ports are 10 Gigabits.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface type slot/port**
3. switch(config-if)# **speed speed**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface type slot/port</b>	Enters interface configuration mode for the specified interface. This interface must have a 1-Gigabit Ethernet SFP transceiver inserted into it.
<b>Step 3</b>	switch(config-if)# <b>speed speed</b>	Sets the speed for a physical Ethernet interface.

The following example shows how to set the speed for a 1-Gigabit Ethernet port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# speed 1000
```

## Disabling Link Negotiation

You can disable link negotiation using the **no negotiate auto** command. By default, auto-negotiation is enabled on 1-Gigabit ports and disabled on 10-Gigabit ports.

This command is equivalent to the Cisco IOS **speed non-negotiate** command.

**Note**

The auto-negotiation configuration is not applicable on 10-Gigabit Ethernet ports. When auto-negotiation is configured on a 10-Gigabit port, the following error message is displayed:

```
ERROR: Ethernet1/40: Configuration does not match the port capability
```

**SUMMARY STEPS**

1. switch# **configure terminal**
2. switch(config)# **interface ethernet slot/port**
3. switch(config-if)# **no negotiate auto**
4. (Optional) switch(config-if)# **negotiate auto**

**DETAILED STEPS**

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface ethernet slot/port</b>	Selects the interface and enters interface mode.
<b>Step 3</b>	switch(config-if)# <b>no negotiate auto</b>	Disables link negotiation on the selected Ethernet interface (1-Gigabit port).
<b>Step 4</b>	switch(config-if)# <b>negotiate auto</b>	(Optional) Enables link negotiation on the selected Ethernet interface. The default for 1-Gigabit Ethernet ports is enabled. <b>Note</b> This command is not applicable for 10GBASE-T ports. It should not be used on 10-GBASE-T ports.

This example shows how to disable auto-negotiation on a specified Ethernet interface (1-Gigabit port):

```
switch# configure terminal
switch(config)# interface ethernet 1/1
switch(config-if)# no negotiate auto
switch(config-if)#
```

This example shows how to enable auto-negotiation on a specified Ethernet interface (1-Gigabit port):

```
switch# configure terminal
switch(config)# interface ethernet 1/5
switch(config-if)# negotiate auto
switch(config-if)#
```

## Configuring the CDP Characteristics

You can configure the frequency of Cisco Discovery Protocol (CDP) updates, the amount of time to hold the information before discarding it, and whether or not to send Version-2 advertisements.

## SUMMARY STEPS

1. switch# **configure terminal**
2. (Optional) switch(config)# **[no] cdp advertise {v1 | v2 }**
3. (Optional) switch(config)# **[no] cdp format device-id {mac-address | serial-number | system-name}**
4. (Optional) switch(config)# **[no] cdp holdtime seconds**
5. (Optional) switch(config)# **[no] cdp timer seconds**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>[no] cdp advertise {v1   v2 }</b>	(Optional) Configures the version to use to send CDP advertisements. Version-2 is the default state.  Use the <b>no</b> form of the command to return to its default setting.
<b>Step 3</b>	switch(config)# <b>[no] cdp format device-id {mac-address   serial-number   system-name}</b>	(Optional) Configures the format of the CDP device ID. The default is the system name, which can be expressed as a fully qualified domain name.  Use the <b>no</b> form of the command to return to its default setting.
<b>Step 4</b>	switch(config)# <b>[no] cdp holdtime seconds</b>	(Optional) Specifies the amount of time a receiving device should hold the information sent by your device before discarding it. The range is 10 to 255 seconds; the default is 180 seconds.  Use the <b>no</b> form of the command to return to its default setting.
<b>Step 5</b>	switch(config)# <b>[no] cdp timer seconds</b>	(Optional) Sets the transmission frequency of CDP updates in seconds. The range is 5 to 254; the default is 60 seconds.  Use the <b>no</b> form of the command to return to its default setting.

This example shows how to configure CDP characteristics:

```
switch# configure terminal
switch(config)# cdp timer 50
switch(config)# cdp holdtime 120
switch(config)# cdp advertise v2
```

## Enabling or Disabling CDP

You can enable or disable CDP for Ethernet interfaces. This protocol works only when you have it enabled on both interfaces on the same link.

**SUMMARY STEPS**

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot/port*
3. switch(config-if)# **cdp enable**
4. switch(config-if)# **no cdp enable**

**DETAILED STEPS**

	<b>Command or Action</b>	<b>Purpose</b>
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface</b> <i>type slot/port</i>	Enters interface configuration mode for the specified interface.
<b>Step 3</b>	switch(config-if)# <b>cdp enable</b>	Enables CDP for the interface.  To work correctly, this parameter must be enabled for both interfaces on the same link.
<b>Step 4</b>	switch(config-if)# <b>no cdp enable</b>	Disables CDP for the interface.

This example shows how to enable CDP for an Ethernet port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# cdp enable
```

This command can only be applied to a physical Ethernet interface.

**Enabling the Error-Disabled Detection**

You can enable error-disable (err-disabled) detection in an application. As a result, when a cause is detected on an interface, the interface is placed in an err-disabled state, which is an operational state that is similar to the link-down state.

**Note**

Base ports in Cisco Nexus 5500 never get error disabled due to pause rate-limit like in the Cisco Nexus 5020 or 5010 switch.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **errdisable detect cause** {all | link-flap | loopback}
3. switch(config)# **shutdown**
4. switch(config)# **no shutdown**
5. switch(config)# **show interface status err-disabled**
6. (Optional) switch(config)# **copy running-config startup-config**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>errdisable detect cause</b> {all   link-flap   loopback}	Specifies a condition under which to place the interface in an err-disabled state. The default is enabled.
<b>Step 3</b>	switch(config)# <b>shutdown</b>	Brings the interface down administratively. To manually recover the interface from the err-disabled state, enter this command first.
<b>Step 4</b>	switch(config)# <b>no shutdown</b>	Brings the interface up administratively and enables the interface to recover manually from the err-disabled state.
<b>Step 5</b>	switch(config)# <b>show interface status err-disabled</b>	Displays information about err-disabled interfaces.
<b>Step 6</b>	switch(config)# <b>copy running-config startup-config</b>	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to enable the err-disabled detection in all cases:

```
switch# configure terminal
switch(config)# errdisable detect cause all
switch(config)# shutdown
switch(config)# no shutdown
switch(config)# show interface status err-disabled
switch(config)# copy running-config startup-config
```

## Enabling the Error-Disabled Recovery

You can specify the application to bring the interface out of the error-disabled (err-disabled) state and retry coming up. It retries after 300 seconds, unless you configure the recovery timer (see the **errdisable recovery interval** command).

**SUMMARY STEPS**

1. switch# **configure terminal**
2. switch(config)# **errdisable recovery cause** {*all* | *udld* | *bpduguard* | *link-flap* | *failed-port-state* | *pause-rate-limit*}
3. switch(config)# **show interface status err-disabled**
4. (Optional) switch(config)# **copy running-config startup-config**

**DETAILED STEPS**

	<b>Command or Action</b>	<b>Purpose</b>
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>errdisable recovery cause</b> { <i>all</i>   <i>udld</i>   <i>bpduguard</i>   <i>link-flap</i>   <i>failed-port-state</i>   <i>pause-rate-limit</i> }	Specifies a condition under which the interface automatically recovers from the err-disabled state, and the device retries bringing the interface up. The device waits 300 seconds to retry. The default is disabled.
<b>Step 3</b>	switch(config)# <b>show interface status err-disabled</b>	Displays information about err-disabled interfaces.
<b>Step 4</b>	switch(config)# <b>copy running-config startup-config</b>	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to enable err-disabled recovery under all conditions:

```
switch# configure terminal
switch(config)# errdisable recovery cause all
switch(config)# show interface status err-disabled
switch(config)# copy running-config startup-config
```

**Configuring the Error-Disabled Recovery Interval**

You can use this procedure to configure the err-disabled recovery timer value. The range is from 30 to 65535 seconds. The default is 300 seconds.

**SUMMARY STEPS**

1. switch# **configure terminal**
2. switch(config)# **errdisable recovery interval** *interval*
3. switch(config)# **show interface status err-disabled**
4. (Optional) switch(config)# **copy running-config startup-config**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>switch# configure terminal</code>	Enters global configuration mode.
Step 2	<code>switch(config)# errdisable recovery interval <i>interval</i></code>	Specifies the interval for the interface to recover from the err-disabled state. The range is from 30 to 65535 seconds. The default is 300 seconds.
Step 3	<code>switch(config)# show interface status err-disabled</code>	Displays information about err-disabled interfaces.
Step 4	<code>switch(config)# copy running-config startup-config</code>	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to enable err-disabled recovery under all conditions:

```
switch# configure terminal
switch(config)# errdisable recovery interval 32
switch(config)# show interface status err-disabled
switch(config)# copy running-config startup-config
```

## Port Profiles

### Creating a Port Profile

You can create a port profile on the switch. Each port profile must have a unique name across interface types and the network.

## SUMMARY STEPS

1. `configure terminal`
2. `port-profile [type {ethernet | interface-vlan | port channel}] name`
3. `exit`
4. (Optional) `show port-profile`
5. (Optional) `copy running-config startup-config`

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>configure terminal</code>  <b>Example:</b> <code>switch# configure terminal</code> <code>switch(config)#</code>	Enters configuration mode.

	Command or Action	Purpose
<b>Step 2</b>	<b>port-profile</b> [ <b>type</b> { <b>ethernet</b>   <b>interface-vlan</b>   <b>port channel</b> }] <i>name</i>  <b>Example:</b> switch(config)# port-profile type ethernet test switch(config-port-prof)#	Creates and names a port profile for the specified type of interface and enters the port profile configuration mode.
<b>Step 3</b>	<b>exit</b>  <b>Example:</b> switch(config-port-prof)# exit switch(config)#	Exits port profile configuration mode.
<b>Step 4</b>	<b>show port-profile</b>  <b>Example:</b> switch(config)# show port-profile <i>name</i>	(Optional) Displays the port profile configuration.
<b>Step 5</b>	<b>copy running-config startup-config</b>  <b>Example:</b> switch(config)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

This example shows how to create a port profile named test for Ethernet interfaces:

```
switch# configure terminal
switch(config)# port-profile type ethernet test
switch(config-port-prof)#
```

This example shows how to add the interface commands to a port profile named ppEth configured for Ethernet interfaces:

```
switch# configure terminal
switch(config)# port-profile ppEth
switch(config-port-prof)# switchport mode trunk
switch(config-port-prof)# switchport trunk allowed vlan 300-400
switch(config-port-prof)# flowcontrol receive on
switch(config-port-prof)# speed 10000
switch(config-port-prof)#
```

## Modifying a Port Profile

You can modify a port profile in port-profile configuration mode.

You can remove commands from a port profile using the **no** form of the command. When you remove a command from the port profile, the corresponding command is removed from the interface that is attached to the port profile.

## SUMMARY STEPS

1. **configure terminal**
2. **port-profile** [type {ethernet | interface-vlan | port channel}] *name*
3. **exit**
4. (Optional) **show port-profile**
5. (Optional) **copy running-config startup-config**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>configure terminal</b>  <b>Example:</b> switch# configure terminal switch(config)#	Enters configuration mode.
<b>Step 2</b>	<b>port-profile</b> [type {ethernet   interface-vlan   port channel}] <i>name</i>  <b>Example:</b> switch(config)# port-profile type ethernet test switch(config-port-prof)#	Enters the port profile configuration mode for the specified port profile and allows you to add or remove configurations to the profile.
<b>Step 3</b>	<b>exit</b>  <b>Example:</b> switch(config-port-prof)# exit switch(config)#	Exits the port profile configuration mode.
<b>Step 4</b>	<b>show port-profile</b>  <b>Example:</b> switch(config)# show port-profile <i>name</i>	(Optional) Displays the port profile configuration.
<b>Step 5</b>	<b>copy running-config startup-config</b>  <b>Example:</b> switch(config)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

This example shows how to remove commands from the port profile named ppEth configured for an Ethernet interface:

```
switch# configure terminal
switch(config)# port-profile ppEth
switch(config-port-prof)# switchport mode trunk
switch(config-port-prof)# switchport trunk allowed vlan 300-400
switch(config-port-prof)# flowcontrol receive on
switch(config-port-prof)# no speed 10000
switch(config-port-prof)#
```

## Enabling a Specific Port Profile

### SUMMARY STEPS

1. **configure terminal**
2. **port-profile** [type {ethernet | interface-vlan | port channel}] *name*
3. **state enabled** *name*
4. **exit**
5. (Optional) **show port-profile**
6. (Optional) **copy running-config startup-config**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>configure terminal</b>  <b>Example:</b> switch# configure terminal switch(config)#	Enters configuration mode.
<b>Step 2</b>	<b>port-profile</b> [type {ethernet   interface-vlan   port channel}] <i>name</i>  <b>Example:</b> switch(config)# port-profile type ethernet test switch(config-port-prof)# no shutdown switch(config-port-prof)#	Enters the port profile configuration mode for the specified port profile.
<b>Step 3</b>	<b>state enabled</b> <i>name</i>  <b>Example:</b> switch(config-port-prof)# state enabled switch(config-port-prof)#	Enables the port profile.
<b>Step 4</b>	<b>exit</b>  <b>Example:</b> switch(config-port-prof)# exit switch(config)#	Exits the port profile configuration mode.
<b>Step 5</b>	<b>show port-profile</b>  <b>Example:</b> switch(config)# show port-profile <i>name</i>	(Optional) Displays the port profile configuration.
<b>Step 6</b>	<b>copy running-config startup-config</b>  <b>Example:</b> switch(config)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

This example shows how to enter port profile configuration mode and enable the port profile:

```
switch# configure terminal
switch(config)# port-profile type ethernet test
switch(config-port-prof)# state enabled
switch(config-port-prof)#
```

## Inheriting a Port Profile

You can inherit a port profile onto an existing port profile. The switch supports four levels of inheritance.

### SUMMARY STEPS

1. **configure terminal**
2. **port-profile** *name*
3. **inherit port-profile** *name*
4. **exit**
5. (Optional) **show port-profile**
6. (Optional) **copy running-config startup-config**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>configure terminal</b>  <b>Example:</b> switch# configure terminal switch(config)#	Enters configuration mode.
<b>Step 2</b>	<b>port-profile</b> <i>name</i>  <b>Example:</b> switch(config)# port-profile test switch(config-port-prof)#	Enters port profile configuration mode for the specified port profile.
<b>Step 3</b>	<b>inherit port-profile</b> <i>name</i>  <b>Example:</b> switch(config-port-prof)# inherit port-profile adam switch(config-port-prof)#	Inherits another port profile onto the existing one. The original port profile assumes all the configurations of the inherited port profile.
<b>Step 4</b>	<b>exit</b>  <b>Example:</b> switch(config-port-prof)# exit switch(config)#	Exits the port profile configuration mode.
<b>Step 5</b>	<b>show port-profile</b>  <b>Example:</b> switch(config)# show port-profile <i>name</i>	(Optional) Displays the port profile configuration.

	Command or Action	Purpose
Step 6	<b>copy running-config startup-config</b>  <b>Example:</b> switch(config)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

This example shows how to inherit the port profile named adam onto the port profile named test:

```
switch# configure terminal
switch(config)# port-profile test
switch(config-ppm)# inherit port-profile adam
switch(config-ppm)#
```

This example shows how to add the interface commands to a port profile named ppEth configured for Ethernet interfaces:

```
switch# configure terminal
switch(config)# port-profile ppEth
switch(config-port-prof)# switchport mode trunk
switch(config-port-prof)# switchport trunk allowed vlan 300-400
switch(config-port-prof)# flowcontrol receive on
switch(config-port-prof)# speed 10000
switch(config-port-prof)#
```

This example shows how to inherit a port profile named ppEth configured for Ethernet interfaces into an existing port profile named test:

```
switch# configure terminal
switch(config)# port-profile test
switch(config-port-prof)# inherit port-profile ppEth
switch(config-port-prof)#
```

This example shows how to assign a port profile named ppEth configured for Ethernet interfaces to a range of Ethernet interfaces:

```
switch# configure terminal
switch(config)# interface ethernet 1/2-5
switch(config-if)# inherit port-profile ppEth
switch(config-if)#
```

This example shows how to remove an inherited port profile named ppEth from an existing port profile named test:

```
switch# configure terminal
switch(config)# port-profile test
switch(config-port-prof)# no inherit port-profile ppEth
switch(config-port-prof)#
```

## Removing an Inherited Port Profile

You can remove an inherited port profile.

## SUMMARY STEPS

1. **configure terminal**
2. **port-profile** *name*
3. **no inherit port-profile** *name*
4. **exit**
5. (Optional) **show port-profile**
6. (Optional) **copy running-config startup-config**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>configure terminal</b>  <b>Example:</b> switch# configure terminal switch(config)#	Enters configuration mode.
Step 2	<b>port-profile</b> <i>name</i>  <b>Example:</b> switch(config)# port-profile test switch(config-port-prof)#	Enters port profile configuration mode for the specified port profile.
Step 3	<b>no inherit port-profile</b> <i>name</i>  <b>Example:</b> switch(config-port-prof)# no inherit port-profile adam switch(config-port-prof)#	Removes an inherited port profile from this port profile.
Step 4	<b>exit</b>  <b>Example:</b> switch(config-port-prof)# exit switch(config)#	Exits the port profile configuration mode.
Step 5	<b>show port-profile</b>  <b>Example:</b> switch(config)# show port-profile <i>name</i>	(Optional) Displays the port profile configuration.
Step 6	<b>copy running-config startup-config</b>  <b>Example:</b> switch(config)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

This example shows how to remove the inherited port profile named adam from the port profile named test:

```
switch# configure terminal
switch(config)# port-profile test
```

```
switch(config-ppm)# no inherit port-profile adam
switch(config-ppm)#
```

## Assigning a Port Profile to a Range of Interfaces

You can assign a port profile to an interface or to a range of interfaces. All of the interfaces must be the same type.

### SUMMARY STEPS

1. **switch# configure terminal**
2. **interface** [*ethernet slot/port* | **interface-vlan** *vlan-id* | **port-channel** *number*]
3. **inherit port-profile** *name*
4. **exit**
5. (Optional) **show port-profile** *name*
6. (Optional) **switch(config)# copy running-config startup-config**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>switch# configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	<b>interface</b> [ <i>ethernet slot/port</i>   <b>interface-vlan</b> <i>vlan-id</i>   <b>port-channel</b> <i>number</i> ]	Selects the range of interfaces.
<b>Step 3</b>	<b>inherit port-profile</b> <i>name</i>	Assigns the specified port profile to the selected interfaces.
<b>Step 4</b>	<b>exit</b>	Exits port profile configuration mode.
<b>Step 5</b>	<b>show port-profile</b> <i>name</i>	(Optional) Displays the port profile configuration.
<b>Step 6</b>	<b>switch(config)# copy running-config startup-config</b>	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to assign the port profile named adam to Ethernet interfaces 2/3 to 2/5, 3/2, and 1/20 to 1/25:

```
switch# configure terminal
switch(config)# interface ethernet 2/3 to 2/5, 3/2, and 1/20 to 1/25
switch(config-if)# inherit port-profile adam
switch(config-if)# exit
switch(config)# show port-profile adam
switch(config)# copy running-config startup-config
```

## Removing a Port Profile from a Range of Interfaces

You can remove a port profile from some or all of the interfaces to which you have applied the profile.

**SUMMARY STEPS**

1. `switch# configure terminal`
2. `interface [ethernet slot/port | interface-vlan vlan-id | port-channel number]`
3. `no inherit port-profile name`
4. `exit`
5. (Optional) `show port-profile`
6. (Optional) `switch(config)# copy running-config startup-config`

**DETAILED STEPS**

	Command or Action	Purpose
Step 1	<code>switch# configure terminal</code>	Enters global configuration mode.
Step 2	<code>interface [ethernet slot/port   interface-vlan vlan-id   port-channel number]</code>	Selects the range of interfaces.
Step 3	<code>no inherit port-profile name</code>	Removes the specified port profile from the selected interfaces.
Step 4	<code>exit</code>	Exits port profile configuration mode.
Step 5	<code>show port-profile</code>	(Optional) Displays the port profile configuration.
Step 6	<code>switch(config)# copy running-config startup-config</code>	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to remove the port profile named adam from Ethernet interfaces 1/3-5:

```
switch# configure terminal
switch(config)# interface ethernet 1/3-5
switch(config-if)# no inherit port-profile adam
switch(config-if)# exit
switch(config)# show port-profile
switch(config)# copy running-config startup-config
```

**Configuration Examples for Port Profiles**

The following example shows how to configure a port profile, inherit the port profile on an Ethernet interface, and enabling the port profile.

```
switch(config)#
switch(config)# show running-config interface Ethernet1/14

!Command: show running-config interface Ethernet1/14
!Time: Thu Aug 26 07:01:32 2010

version 5.0(2)N1(1)

interface Ethernet1/14

switch(config)# port-profile type ethernet alpha
```

```

switch(config-port-prof)# switchport mode trunk
switch(config-port-prof)# switchport trunk allowed vlan 10-15
switch(config-port-prof)#
switch(config-port-prof)# show running-config port-profile alpha

!Command: show running-config port-profile alpha
!Time: Thu Aug 26 07:02:29 2010

version 5.0(2)N1(1)
port-profile type ethernet alpha
    switchport mode trunk
    switchport trunk allowed vlan 10-15

switch(config-port-prof)# int eth 1/14
switch(config-if)# inherit port-profile alpha
switch(config-if)#
switch(config-if)# port-profile type ethernet alpha
switch(config-port-prof)# state enabled
switch(config-port-prof)#
switch(config-port-prof)# sh running-config interface ethernet 1/14

!Command: show running-config interface Ethernet1/14
!Time: Thu Aug 26 07:03:17 2010

version 5.0(2)N1(1)

interface Ethernet1/14
    inherit port-profile alpha

switch(config-port-prof)# sh running-config interface ethernet 1/14 expand-port-profile

!Command: show running-config interface Ethernet1/14 expand-port-profile
!Time: Thu Aug 26 07:03:21 2010

version 5.0(2)N1(1)

interface Ethernet1/14
    switchport mode trunk
    switchport trunk allowed vlan 10-15

switch(config-port-prof)#

```

## Configuring the Debounce Timer

You can enable the debounce timer for Ethernet ports by specifying a debounce time (in milliseconds) or disable the timer by specifying a debounce time of 0.

You can show the debounce times for all of the Ethernet ports by using the **show interface debounce** command.

To enable or disable the debounce timer, perform this task:

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot/port*
3. switch(config-if)# **link debounce time** *milliseconds*

## DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	switch(config)# <b>interface</b> <i>type slot/port</i>	Enters interface configuration mode for the specified interface.
Step 3	switch(config-if)# <b>link debounce time</b> <i>milliseconds</i>	Enables the debounce timer for the amount of time (1 to 5000 milliseconds) specified.  Disables the debounce timer if you specify 0 milliseconds.

This example shows how to enable the debounce timer and set the debounce time to 1000 milliseconds for an Ethernet interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# link debounce time 1000
```

This example shows how to disable the debounce timer for an Ethernet interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# link debounce time 0
```

## Configuring the Description Parameter

You can provide textual interface descriptions for the Ethernet ports.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot/port*
3. switch(config-if)# **description** *test*

## DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	switch(config)# <b>interface</b> <i>type slot/port</i>	Enters interface configuration mode for the specified interface.
Step 3	switch(config-if)# <b>description</b> <i>test</i>	Specifies the description for the interface.

This example shows how to set the interface description to Server 3 interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/3
switch(config-if)# description Server 3 Interface
```

## Disabling and Restarting Ethernet Interfaces

You can shut down and restart an Ethernet interface. This action disables all of the interface functions and marks the interface as being down on all monitoring displays. This information is communicated to other network servers through all dynamic routing protocols. When shut down, the interface is not included in any routing updates.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot/port*
3. switch(config-if)# **shutdown**
4. switch(config-if)# **no shutdown**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface</b> <i>type slot/port</i>	Enters interface configuration mode for the specified interface.
<b>Step 3</b>	switch(config-if)# <b>shutdown</b>	Disables the interface.
<b>Step 4</b>	switch(config-if)# <b>no shutdown</b>	Restarts the interface.

This example shows how to disable an Ethernet port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# shutdown
```

This example shows how to restart an Ethernet interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# no shutdown
```

## Displaying Interface Information

To view configuration information about the defined interfaces, perform one of these tasks:

Command	Purpose
switch# <b>show interface</b> <i>type slot/port</i>	Displays the detailed configuration of the specified interface.
switch# <b>show interface</b> <i>type slot/port capabilities</i>	Displays detailed information about the capabilities of the specified interface. This option is available only for physical interfaces.
switch# <b>show interface</b> <i>type slot/port transceiver</i>	Displays detailed information about the transceiver connected to the specified interface. This option is available only for physical interfaces.
switch# <b>show interface brief</b>	Displays the status of all interfaces.
switch# <b>show interface flowcontrol</b>	Displays the detailed listing of the flow control settings on all interfaces.
switch# <b>show interface debounce</b>	Displays the debounce status of all interfaces.
<b>show port--profile</b>	Displays information about the port profiles.

The **show interface** command is invoked from EXEC mode and displays the interface configurations. Without any arguments, this command displays the information for all the configured interfaces in the switch.

This example shows how to display the physical Ethernet interface:

```
switch# show interface ethernet 1/1
Ethernet1/1 is up
Hardware is 1000/10000 Ethernet, address is 000d.eca3.5f08 (bia 000d.eca3.5f08)
MTU 1500 bytes, BW 10000000 Kbit, DLY 10 usec,
    reliability 255/255, txload 190/255, rxload 192/255
Encapsulation ARPA
Port mode is trunk
full-duplex, 10 Gb/s, media type is 1/10g
Input flow-control is off, output flow-control is off
Auto-mdix is turned on
Rate mode is dedicated
Switchport monitor is off
Last clearing of "show interface" counters never
5 minute input rate 942201806 bytes/sec, 14721892 packets/sec
5 minute output rate 935840313 bytes/sec, 14622492 packets/sec
Rx
 129141483840 input packets 0 unicast packets 129141483847 multicast packets
 0 broadcast packets 0 jumbo packets 0 storm suppression packets
8265054965824 bytes
 0 No buffer 0 runt 0 Overrun
 0 crc 0 Ignored 0 Bad etype drop
 0 Bad proto drop
Tx
 119038487241 output packets 119038487245 multicast packets
 0 broadcast packets 0 jumbo packets
7618463256471 bytes
 0 output CRC 0 ecc
 0 underrun 0 if down drop      0 output error 0 collision 0 deferred
 0 late collision 0 lost carrier 0 no carrier
 0 babble
 0 Rx pause 8031547972 Tx pause 0 reset
```

This example shows how to display the physical Ethernet capabilities:

```
switch# show interface ethernet 1/1 capabilities
Ethernet1/1
  Model:                734510033
  Type:                 10Gbase-(unknown)
  Speed:                1000,10000
  Duplex:               full
  Trunk encap. type:    802.1Q
  Channel:              yes
  Broadcast suppression: percentage(0-100)
  Flowcontrol:          rx-(off/on),tx-(off/on)
  Rate mode:            none
  QOS scheduling:       rx-(6q1t),tx-(1p6q0t)
  CoS rewrite:          no
  ToS rewrite:          no
  SPAN:                 yes
  UDLD:                 yes
  Link Debounce:        yes
  Link Debounce Time:  yes
  MDIX:                 no
  FEX Fabric:           yes
```

This example shows how to display the physical Ethernet transceiver:

```
switch# show interface ethernet 1/1 transceiver
Ethernet1/1
  sfp is present
  name is CISCO-EXCELIGHT
  part number is SPP5101SR-C1
  revision is A
  serial number is ECL120901AV
  nominal bitrate is 10300 Mbits/sec
  Link length supported for 50/125mm fiber is 82 m(s)
  Link length supported for 62.5/125mm fiber is 26 m(s)
  cisco id is --
  cisco extended id number is 4
```

This example shows how to display a brief interface status (some of the output has been removed for brevity):

```
switch# show interface brief
```

```
-----
Ethernet      VLAN   Type Mode   Status Reason                               Speed   Port
Interface                                           Ch #
-----
Eth1/1        200   eth trunk up      none                               10G(D)  --
Eth1/2         1     eth trunk up      none                               10G(D)  --
Eth1/3        300   eth access down   SFP not inserted                 10G(D)  --
Eth1/4        300   eth access down   SFP not inserted                 10G(D)  --
Eth1/5        300   eth access down   Link not connected                1000(D) --
Eth1/6        20    eth access down   Link not connected                10G(D)  --
Eth1/7        300   eth access down   SFP not inserted                 10G(D)  --
...

```

This example shows how to display the link debounce status (some of the output has been removed for brevity):

```
switch# show interface debounce
```

```
-----
Port          Debounce time  Value(ms)
-----
...
Eth1/1        enable         100
Eth1/2        enable         100
Eth1/3        enable         100
...

```

This example shows how to display the CDP neighbors:



**Note** The default device ID field for CDP advertisement is the hostname and serial number, as in the example above.

```
switch# show cdp neighbors
Capability Codes: R - Router, T - Trans-Bridge, B - Source-Route-Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater,
                  V - VoIP-Phone, D - Remotely-Managed-Device,
                  s - Supports-STP-Dispute
Device ID         Local Intrfce  Hldtme  Capability  Platform  Port ID
dl3-dist-1       mgmt0         148     S I         WS-C2960-24TC  Fas0/9
n5k (FLC12080012) Eth1/5        8       S I s       N5K-C5020P-BA  Eth1/5
```

## Default Physical Ethernet Settings

The following table lists the default settings for all physical Ethernet interfaces:

Parameter	Default Setting
Debounce	Enable, 100 milliseconds
Duplex	Auto (full-duplex)
Encapsulation	ARPA
MTU <sup>1</sup>	1500 bytes
Port Mode	Access
Speed	Auto (10000)

<sup>1</sup> MTU cannot be changed per-physical Ethernet interface. You modify MTU by selecting maps of QoS classes.





## Configuring VLANs

---

This chapter contains the following sections:

- [Information About VLANs, page 39](#)
- [Configuring a VLAN, page 43](#)

### Information About VLANs

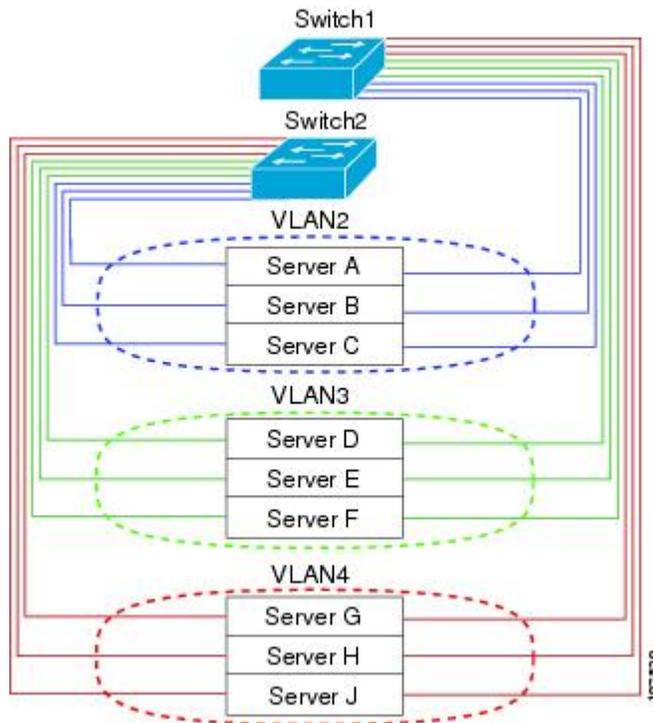
#### Understanding VLANs

A VLAN is a group of end stations in a switched network that is logically segmented by function, project team, or application, without the limitation 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 port can belong to a VLAN; all unicast, broadcast, and multicast packets are forwarded and flooded only to end stations in that VLAN. Each VLAN is considered a logical network. If a packet destination address does not belong to the VLAN, it must be forwarded through a router.

The following figure shows VLANs as logical networks. In this diagram, 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 yet 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. To disable the VLAN use the **shutdown** command. Additionally, you can configure VLANs to be in the active state (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.



**Note**

The VLAN Trunking Protocol (VTP) mode is OFF. VTP BPDUs are dropped on all interfaces of the switch. This process has the effect of partitioning VTP domains if other switches have VTP turned on.

A VLAN can also be configured as a switched virtual interface (SVI). In this case, the switch ports in the VLAN are represented by a virtual interface to a routing or bridging system. The SVI can be configured for routing, in which case it supports Layer 3 protocols for processing packets from all switch ports associated with the VLAN, or for in-band management of the switch.

## Understanding VLAN Ranges

The Cisco Nexus device supports VLAN numbers 1 to 4094 in accordance with the IEEE 802.1Q standard. These VLANs are organized into ranges. The switch is physically limited in the number of VLANs it can

support. The hardware also shares this available range with its VSANs. For information about VLAN and VSAN configuration limits, see the configuration limits documentation for your device.

The following table describes the details of the VLAN ranges.

**Table 4: 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.
1006—4094	Extended	You can create, name, and use these VLANs. You cannot change the following parameters: <ul style="list-style-type: none"> <li>• State is always active.</li> <li>• VLAN is always enabled. You cannot shut down these VLANs.</li> </ul>
3968—4049 and 4094	Internally allocated	These 82 VLANs, plus VLAN 4094, are allocated for internal use. You cannot create, delete, or modify any VLANs within the block reserved for internal use.



**Note** You cannot configure the internally allocated VLANs (reserved VLANs).



**Note** VLANs 3968 to 4049 and 4094 are reserved for internal use; these VLANs cannot be changed or used.

Cisco NX-OS allocates a group of 82 VLAN numbers for those features, such as multicast and diagnostics, that need to use internal VLANs for their operation. By default, the system allocates VLANs numbered 3968 to 4049 for internal use. VLAN 4094 is also reserved for internal use by the switch.

You cannot use, modify, or delete any of the VLANs in the reserved group. You can display the VLANs that are allocated internally and their associated use.

## Creating, Deleting, and Modifying VLANs

VLANs are numbered from 1 to 4094. All configured ports belong to the default VLAN when you first bring up the switch. The default VLAN (VLAN1) uses only default values. 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 as well as 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 switch goes into the VLAN submode but does not create the same VLAN again.

Newly created VLANs remain unused until 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
- Shutdown or not shutdown

When you delete a specified VLAN, the ports associated to that VLAN are shut down and no traffic flows. However, the system retains all the VLAN-to-port mapping for that VLAN, and when you reenables, or recreates, the specified VLAN, the system automatically reinstates all the original ports to that VLAN.

**Note**

---

Commands entered in the VLAN configuration submode are immediately executed.

VLANs 3968 to 4049 and 4094 are reserved for internal use; these VLANs cannot be changed or used.

---

## About the VLAN Trunking Protocol

VLAN Trunking Protocol (VTP) is a distributed VLAN database management protocol that synchronizes the VTP VLAN database across domains. A VTP domain includes one or more network switches that share the same VTP domain name and are connected with trunk interfaces.

The following are the different VTP modes:

- Server mode—Allows users to perform configurations, manage the VLAN database version, and store the VLAN database.
- Client mode—Does not allow users to perform configurations and relies on other switches in the domain to provide configuration information.
- Off mode—Allows users to access the VLAN database (VTP is enabled) but does not participate in VTP.
- Transparent mode—Does not participate in VTP, uses local configuration, and relays VTP packets to other forward ports. VLAN changes affect only the local switch. A VTP transparent network switch does not advertise its VLAN configuration and does not synchronize its VLAN configuration based on received advertisements.

## Guidelines and Limitations for VTP

VTP has the following configuration guidelines and limitations:

- When a switch is configured as a VTP client, you cannot create VLANs on the switch in the range of 1 to 1005.
- 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.
- If you enable VTP, you must configure either version 1 or version 2. On the Cisco Nexus device, 512 VLANs are supported. If these switches are in a distribution network with other switches, the limit remains the same.

On the Cisco Nexus device, 512 VLANs are supported. If these switches are in a distribution network with other switches, the VLAN limit for the VTP domain is 512. If a Cisco Nexus device client/server receives additional VLANs from a VTP server, they transition to transparent mode.

- If **system vlan long-name** knob is enabled, then VTP configurations will come up in OFF mode and users can change the mode to Transparent. However, changing the mode to Server or Client is not allowed.
- The **show running-configuration** command does not show VLAN or VTP configuration information for VLANs 1 to 1000.
- When deployed with vPC, both vPC switches must be configured identically. vPC performs a Type 2 consistency check for VTP configuration parameters.
- VTP advertisements are not sent out on Cisco Nexus Fabric Extender ports.
- Private VLANs (PVLANS) are supported only when the switch is in transparent mode.
- If you are using VTP in a Token Ring environment, you must use version 2.
- When a switch is configured in VTP client or server mode, VLANs 1002 to 1005 are reserved VLANs.
- VTP pruning is not supported.
- You must enter the **copy running-config startup-config** command followed by a reload after changing a reserved VLAN range. For example:

```
switch(config)# system vlan 2000 reserve
This will delete all configs on vlans 2000-2081. Continue anyway? (y/n) [no] y
```

After the switch reload, VLANs 2000 to 2081 are reserved for internal use, which requires that you enter the **copy running-config startup-config** command before the switch reload. Creating VLANs within this range is not allowed.

## Configuring a VLAN

### Creating and Deleting a VLAN

You can create or delete all VLANs except the default VLAN and those VLANs that are internally allocated for use by the switch. Once a VLAN is created, it is automatically in the active state.

**Note**

When you delete a VLAN, ports associated to that VLAN shut down. The traffic does not flow and the packets are dropped.

**SUMMARY STEPS**

1. switch# **configure terminal**
2. switch(config)# **vlan** {vlan-id | vlan-range}
3. switch(config-vlan)# **no vlan** {vlan-id | vlan-range}

**DETAILED STEPS**

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>vlan</b> {vlan-id   vlan-range}	Creates a VLAN or a range of VLANs.  If you enter a number that is already assigned to a VLAN, the switch moves 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 <i>only</i> those VLANs outside the range. The range is from 2 to 4094; VLAN1 is the default VLAN and cannot be created or deleted. You cannot create or delete those VLANs that are reserved for internal use.
<b>Step 3</b>	switch(config-vlan)# <b>no vlan</b> {vlan-id   vlan-range}	Deletes the specified VLAN or range of VLANs and removes you from the VLAN configuration submode. You cannot delete VLAN1 or the internally allocated VLANs.

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

```
switch# configure terminal
switch(config)# vlan 15-20
```

**Note**

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

## Changing the Range of Reserved VLANs

To change the range of reserved VLANs, you must be in global configuration mode. After entering this command, you must do the following tasks:

- Enter the **copy running-config startup-config** command
- Reload the device

## SUMMARY STEPS

1. `config t`
2. `system vlan start-vlan reserve`
3. `copy running-config startup-config`
4. `reload`
5. (Optional) `show system vlan reserved`

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>config t</b>  <b>Example:</b> <pre>switch# config t switch(config)#</pre>	Enters global configuration mode.
Step 2	<b>system vlan start-vlan reserve</b>  <b>Example:</b> <pre>switch(config)# system vlan 3968 reserve</pre>	<p>Allows you to change the reserved VLAN range by specifying the starting VLAN ID for your desired range.</p> <p>You can change the reserved VLANs to any other 82 contiguous VLAN ranges. When you reserve such a range, it frees up the range of VLANs that were allocated for internal use by default, and all of those VLANs are available for user configuration except for VLAN 4094.</p> <p><b>Note</b> To return to the default range of reserved VLANs (3968-4049 and 4094), you must enter the <b>no system vlan start-vlan reserve</b> command.</p>
Step 3	<b>copy running-config startup-config</b>  <b>Example:</b> <pre>switch(config)# copy running-config startup-config</pre>	<p>Copies the running configuration to the startup configuration.</p> <p><b>Note</b> You must enter this command if you change the reserved block.</p>
Step 4	<b>reload</b>  <b>Example:</b> <pre>switch(config)# reload</pre>	<p>Reloads the software, and modifications to VLAN ranges become effective.</p> <p>For more details about this command, see the <i>Cisco Nexus 7000 Series NX-OS Fundamentals Configuration Guide, Release 6.x</i>.</p>
Step 5	<b>show system vlan reserved</b>  <b>Example:</b> <pre>switch(config)# show system vlan reserved</pre>	<p>(Optional)</p> <p>Displays the configured changes to the VLAN range.</p>

This example shows how to change the range of reserved VLANs:

```
switch# configuration terminal
switch(config)# system vlan 1006 reserve
This will delete all configs on vlans 1006-1087. Continue anyway? (y/n) [no] yes
Note: After switch reload, VLANs 1006-1087 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)# copy running-config startup-config
switch(config)# reload
switch(config)# show system vlan reserved

```



**Note** You must reload the device for this change to take effect.

## Configuring a VLAN

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

- Name



**Note** VLAN name can be either a short name (up to 32 characters) or long name (up to 128 characters). To configure VLAN long-name of up to 128 characters, you must enable **system vlan long-name** command.

- 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. switch# **configure terminal**
2. switch(config)# **vlan** {vlan-id | vlan-range}
3. switch(config-vlan)# **name** vlan-name
4. switch(config-vlan)# **state** {active | suspend}
5. (Optional) switch(config-vlan)# **no shutdown**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>vlan</b> {vlan-id   vlan-range}	Enters VLAN configuration submode. If the VLAN does not exist, the system first creates the specified VLAN.
<b>Step 3</b>	switch(config-vlan)# <b>name</b> vlan-name	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 represents four numeric digits (including leading zeroes) equal to the VLAN ID number.

	Command or Action	Purpose
<b>Step 4</b>	switch(config-vlan)# <b>state</b> { <b>active</b>   <b>suspend</b> }	Sets the state of the VLAN to active or suspend. While the VLAN state is suspended, the ports associated with this VLAN are shut down, 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 4094.
<b>Step 5</b>	switch(config-vlan)# <b>no shutdown</b>	(Optional) Enables the VLAN. The default value is <b>no shutdown</b> (or enabled). You cannot shut down the default VLAN, VLAN1, or VLANs 1006 to 4094.

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

```
switch# configure terminal
switch(config)# vlan 5
switch(config-vlan)# name accounting
switch(config-vlan)# state active
switch(config-vlan)# no shutdown
```

## Adding Ports to a VLAN

After you have completed the configuration of a VLAN, assign ports to it.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** {*ethernet slot/port* | **port-channel number**}
3. switch(config-if)# **switchport access vlan** *vlan-id*

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface</b> { <i>ethernet slot/port</i>   <b>port-channel number</b> }	Specifies the interface to configure, and enters the interface configuration mode. The interface can be a physical Ethernet port or an EtherChannel.
<b>Step 3</b>	switch(config-if)# <b>switchport access vlan</b> <i>vlan-id</i>	Sets the access mode of the interface to the specified VLAN.

This example shows how to configure an Ethernet interface to join VLAN 5:

```
switch# configure terminal
switch(config)# interface ethernet 1/13
switch(config-if)# switchport access vlan 5
```

## Configuring VTP

You can configure VTP in the client or server mode on Cisco Nexus devices.

You can enable VTP and then configure the VTP mode (server [default], client, transparent, or off). If you enable VTP, you must configure either version 1 or version 2. If you are using VTP in a Token Ring environment, you must use version 2.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **feature vtp**
3. switch(config)# **vtp domain** *domain-name*
4. switch(config)# **vtp version** {1 | 2}
5. switch(config)# **vtp file** *file-name*
6. switch(config)# **vtp password** *password-value*
7. switch(config)# **exit**
8. (Optional) switch# **show vtp status**
9. (Optional) switch# **show vtp counters**
10. (Optional) switch# **show vtp interface**
11. (Optional) switch# **show vtp password**
12. (Optional) switch# **copy running-config startup-config**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>feature vtp</b>	Enables VTP on the device. The default is disabled.
<b>Step 3</b>	switch(config)# <b>vtp domain</b> <i>domain-name</i>	Specifies the name of the VTP domain that you want this device to join. The default is blank.
<b>Step 4</b>	switch(config)# <b>vtp version</b> {1   2}	Sets the VTP version that you want to use. The default is version 1.
<b>Step 5</b>	switch(config)# <b>vtp file</b> <i>file-name</i>	Specifies the ASCII filename of the IFS file system file where the VTP configuration is stored.
<b>Step 6</b>	switch(config)# <b>vtp password</b> <i>password-value</i>	Specifies the password for the VTP administrative domain.
<b>Step 7</b>	switch(config)# <b>exit</b>	Exits the configuration submode.
<b>Step 8</b>	switch# <b>show vtp status</b>	(Optional) Displays information about the VTP configuration on the device, such as the version, mode, and revision number.

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

This example shows the VTP status and that the switch is capable of supporting Version 2 and that the switch is running Version 1:

```
switch(config)# show vtp status
VTP Status Information
-----
VTP Version                : 2 (capable)
Configuration Revision     : 0
Maximum VLANs supported locally : 1005
Number of existing VLANs   : 502
VTP Operating Mode        : Transparent
VTP Domain Name           :
VTP Pruning Mode          : Disabled (Operationally Disabled)
VTP V2 Mode                : Disabled
VTP Traps Generation      : Disabled
MD5 Digest                 : 0xF5 0xF1 0xEC 0xE7 0x29 0x0C 0x2D 0x01
Configuration last modified by 60.10.10.1 at 0-0-00 00:00:00
VTP version running       : 1
```

## Verifying the VLAN Configuration

Use one of the following commands to verify the configuration:

Command	Purpose
switch# <b>show running-config vlan</b> [ <i>vlan_id</i>   <i>vlan_range</i> ]	Displays VLAN information.
switch# <b>show vlan</b> [ <b>brief</b>   <b>id</b> [ <i>vlan_id</i>   <i>vlan_range</i> ]   <b>name name</b>   <b>summary</b> ]	Displays selected configuration information for the defined VLAN(s).
switch# <b>show system vlan reserved</b>	Displays the system reserved VLAN range.





## Configuring Private VLANs

---

This chapter contains the following sections:

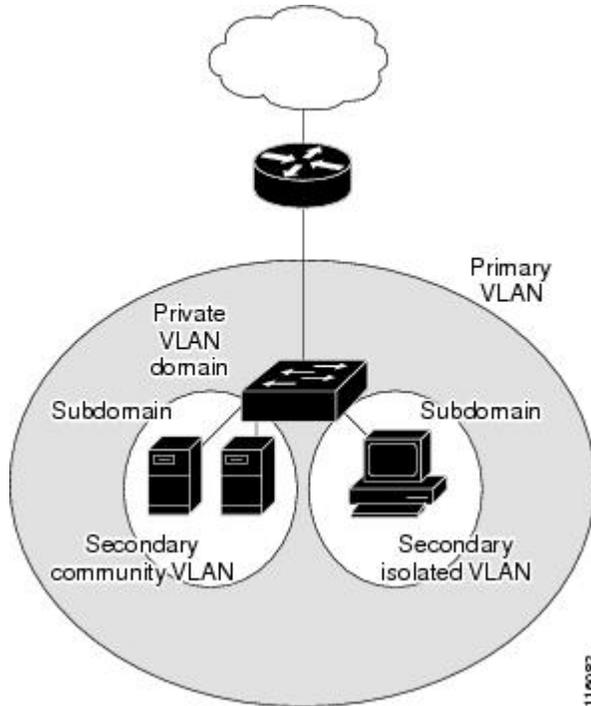
- [Information About Private VLANs, page 51](#)
- [Guidelines and Limitations for Private VLANs, page 56](#)
- [Configuring a Private VLAN, page 57](#)
- [Verifying the Private VLAN Configuration, page 67](#)

### Information About Private VLANs

A private VLAN (PVLAN) partitions the Ethernet broadcast domain of a VLAN into subdomains, allowing you to isolate the ports on the switch from each other. A subdomain consists of a primary VLAN and one or more secondary VLANs (see the following figure). All VLANs in a PVLAN domain share the same primary VLAN. The secondary VLAN ID differentiates one subdomain from another. The secondary VLANs can either be isolated VLANs or community VLANs. A host on an isolated VLAN can communicate only with

the associated promiscuous port in its primary VLAN. Hosts on community VLANs can communicate among themselves and with their associated promiscuous port but not with ports in other community VLANs.

**Figure 3: Private VLAN Domain**



**Note** You must first create the VLAN before you can convert it to a PVLAN, either primary or secondary.

## Primary and Secondary VLANs in Private VLANs

A private VLAN domain has only one primary VLAN. Each port in a private VLAN domain is a member of the primary VLAN; the primary VLAN is the entire private VLAN domain.

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

- Isolated VLANs—Ports within an isolated VLAN cannot communicate directly 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

The three types of PVLAN 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 that are 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 for load-balancing or redundancy purposes. You can also have secondary VLANs that are not associated to any promiscuous port.

A promiscuous port can be configured either as an access port or as a trunk port.

- **Isolated port**—An isolated port is a host port that belongs to an isolated secondary VLAN. This port has complete isolation from other ports within the same PVLAN domain, except that it can communicate with associated promiscuous ports. PVLANS 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. Each port is completely isolated from all other ports in the isolated VLAN.

An isolated port can be configured as either an access port or a trunk port.

- **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 from all other interfaces in other communities and from all isolated ports within the PVLAN domain.

A community port must be configured as an access port. A community VLAN must not be enabled on an isolated trunk.




---

**Note** A trunk port on the Fabric Extender (FEX) can be either a FEX trunk port or a FEX isolated trunk port.

---




---

**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 switch through a trunk interface.

---

## Primary, Isolated, and Community Private VLANs

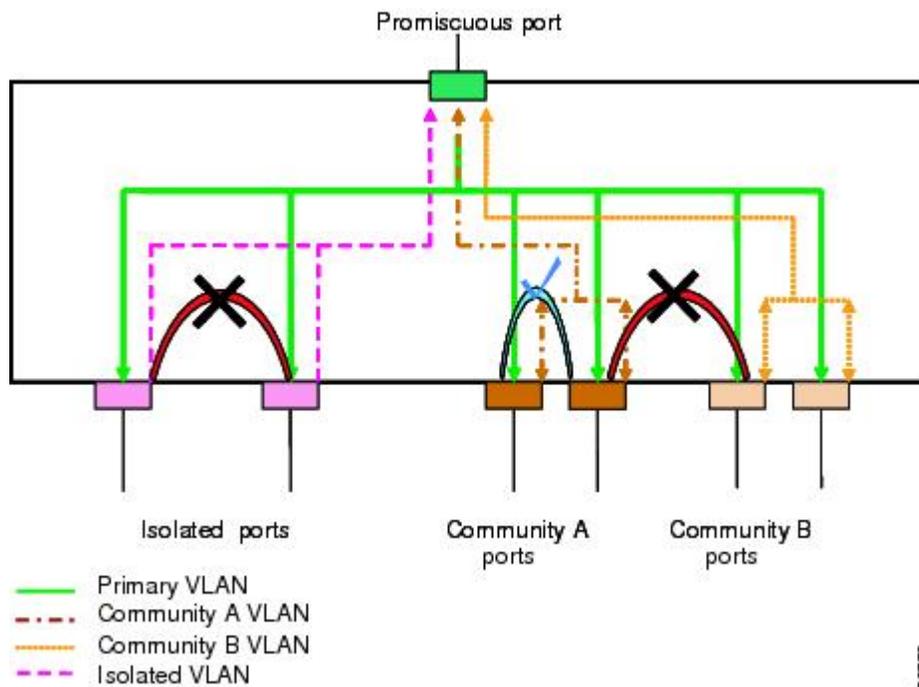
Primary VLANs and the two types of secondary VLANs (isolated and community) have these characteristics:

- **Primary VLAN**— The primary VLAN carries traffic from the promiscuous ports to the host ports, both isolated and community, 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. You can configure only one isolated VLAN in a PVLAN domain. An isolated VLAN can have several isolated ports. 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 and to other host ports in the same community. You can configure multiple community VLANs in a PVLAN 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.

The following figure shows the traffic flows within a PVLAN, along with the types of VLANs and types of ports.

**Figure 4: Private VLAN Traffic Flows**



**Note**

The PVLAN traffic flows are unidirectional from the host ports to the promiscuous ports. Traffic received on primary VLAN enforces no separation and forwarding is done as in a normal VLAN.

A promiscuous access port can serve only one primary VLAN and multiple secondary VLANs (community and isolated VLANs). A promiscuous trunk port can carry traffic for several primary VLANs. Multiple secondary VLANs under a given primary VLAN can be mapped to promiscuous trunk ports. With a promiscuous port, you can connect a wide range of devices as access points to a PVLAN. For example, you can use a promiscuous port to monitor or back up all the PVLAN servers from an administration workstation.

In a switched environment, you can assign an individual PVLAN 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.

## Associating Primary and Secondary VLANs

To allow host ports in secondary VLANs to communicate outside the PVLAN, you associate secondary VLANs to the primary VLAN. If the association is not operational, the host ports (community and isolated 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 and be configured as a primary VLAN.
- The secondary VLAN must exist and be configured as either an isolated or community VLAN.

**Note**

Use the **show vlan private-vlan** command to verify that the association is operational. The switch does not display 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. Use the **no private-vlan** command to return the VLAN to the normal mode. All primary and secondary associations on that VLAN are suspended, but the interfaces remain in PVLAN mode. When you convert the VLAN back to PVLAN mode, the original associations are reinstated.

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

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.

## Private VLAN Promiscuous Trunks

A promiscuous trunk port can carry traffic for several primary VLANs. Multiple secondary VLANs under a given primary VLAN can be mapped to a promiscuous trunk port. Traffic on the promiscuous port is received and transmitted with a primary VLAN tag.

## Private VLAN Isolated Trunks

An isolated trunk port can carry traffic for multiple isolated PVLANS. Traffic for a community VLAN is not carried by isolated trunk ports. Traffic on isolated trunk ports is received and transmitted with an isolated VLAN tag. Isolated trunk ports are intended to be connected to host servers.

To support isolated PVLAN ports on a Cisco Nexus Fabric Extender, the Cisco Nexus device must prevent communication between the isolated ports on the FEX; all forwarding occurs through the switch.

**Caution**

You must disable all the FEX isolated trunk ports before configuring PVLANS on the FEX trunk ports. If the FEX isolated trunk ports and the FEX trunk ports are both enabled, unwanted network traffic might occur.

For unicast traffic, you can prevent such a communication without any side effects.

For multicast traffic, the FEX provides replication of the frames. To prevent communication between isolated PVLAN ports on the FEX, the switch prevents multicast frames from being sent back through the fabric ports.

This restriction prevents communication between an isolated VLAN and a promiscuous port on the FEX. However, as host interfaces are not intended to be connected to another switch or router, you cannot enable a promiscuous port on a FEX.

## Broadcast Traffic in Private VLANs

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

- The broadcast traffic flows from a promiscuous port to all ports in the primary VLAN (which includes all the ports in the community and isolated VLANs). 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 an isolated port 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 PVLANS to control access to end stations as follows:

- Configure selected interfaces connected to end stations as isolated ports to prevent any communication. For example, if the end stations are servers, this configuration prevents 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.

## Guidelines and Limitations for Private VLANs

When configuring PVLANS, follow these guidelines:

- You must have already created the VLAN before you can assign the specified VLAN as a private VLAN.
- You must enable PVLANS before the switch can apply the PVLAN functionality.
- You cannot disable PVLANS if the switch has any operational ports in a PVLAN mode.
- Enter the **private-vlan synchronize** command from within the Multiple Spanning Tree (MST) region definition to map the secondary VLANs to the same MST instance as the primary VLAN.
- You must disable all the FEX isolated trunk ports before configuring FEX trunk ports.
- You cannot connect a second switch to a promiscuous or isolated PVLAN trunk. The promiscuous or isolated PVLAN trunk is supported only on host-switch.

# Configuring a Private VLAN

## Enabling Private VLANs

You must enable PVLANs on the switch to use the PVLAN functionality.


**Note**

The PVLAN commands do not appear until you enable the PVLAN feature.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **feature private-vlan**
3. (Optional) switch(config)# **no feature private-vlan**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>feature private-vlan</b>	Enables the PVLAN feature on the switch.
<b>Step 3</b>	switch(config)# <b>no feature private-vlan</b>	(Optional) Disables the PVLAN feature on the switch. <b>Note</b> You cannot disable PVLANs if there are operational ports on the switch that are in PVLAN mode.

This example shows how to enable the PVLAN feature on the switch:

```
switch# configure terminal
switch(config)# feature private-vlan
```

## Configuring a VLAN as a Private VLAN

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

### Before You Begin

Ensure that the PVLAN feature is enabled.

**SUMMARY STEPS**

1. switch# **configure terminal**
2. switch(config)# **vlan** {vlan-id | vlan-range}
3. switch(config-vlan)# **private-vlan** {community | isolated | primary}
4. (Optional) switch(config-vlan)# **no private-vlan** {community | isolated | primary}

**DETAILED STEPS**

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>vlan</b> {vlan-id   vlan-range}	Places you into the VLAN configuration submenu.
<b>Step 3</b>	switch(config-vlan)# <b>private-vlan</b> { <b>community</b>   <b>isolated</b>   <b>primary</b> }	Configures the VLAN as either a community, isolated, or primary PVLAN. In a PVLAN, you must have one primary VLAN. You can have multiple community and isolated VLANs.
<b>Step 4</b>	switch(config-vlan)# <b>no private-vlan</b> { <b>community</b>   <b>isolated</b>   <b>primary</b> }	(Optional) Removes the PVLAN 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.

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

```
switch# configure terminal
switch(config)# vlan 5
switch(config-vlan)# private-vlan primary
```

This example shows how to assign VLAN 100 to a PVLAN as a community VLAN:

```
switch# configure terminal
switch(config)# vlan 100
switch(config-vlan)# private-vlan community
```

This example shows how to assign VLAN 200 to a PVLAN as an isolated VLAN:

```
switch# configure terminal
switch(config)# vlan 200
switch(config-vlan)# private-vlan isolated
```

**Associating Secondary VLANs with a Primary Private VLAN**

When you associate secondary VLANs with a primary VLAN, follow these guidelines:

- 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 VLAN IDs and one isolated VLAN ID.

- Enter a *secondary-vlan-list* or use the **add** keyword with a *secondary-vlan-list* to associate secondary VLANs with a primary VLAN.
- Use 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 VLAN becomes inactive on the port where the association is configured. When you enter the **no private-vlan** command, the VLAN returns to the normal VLAN mode. All primary and secondary associations on that VLAN are suspended, but the interfaces remain in PVLAN mode. If you again convert the specified VLAN to PVLAN mode, the original associations are reinstated.

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

### Before You Begin

Ensure that the PVLAN feature is enabled.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vlan** *primary-vlan-id*
3. switch(config-vlan)# **private-vlan association** {[**add**] *secondary-vlan-list* | **remove** *secondary-vlan-list*}
4. (Optional) switch(config-vlan)# **no private-vlan association**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>vlan</b> <i>primary-vlan-id</i>	Enters the number of the primary VLAN that you are working in for the PVLAN configuration.
<b>Step 3</b>	switch(config-vlan)# <b>private-vlan association</b> {[ <b>add</b> ] <i>secondary-vlan-list</i>   <b>remove</b> <i>secondary-vlan-list</i> }	Associates the secondary VLANs with the primary VLAN. Use the <b>remove</b> keyword with a <i>secondary-vlan-list</i> to clear the association between secondary VLANs and a primary VLAN.
<b>Step 4</b>	switch(config-vlan)# <b>no private-vlan association</b>	(Optional) Removes all associations from the primary VLAN and returns it to normal VLAN mode.

This example shows how to associate community VLANs 100 through 110 and isolated VLAN 200 with primary VLAN 5:

```
switch# configure terminal
switch(config)# vlan 5
switch(config-vlan)# private-vlan association 100-110, 200
```

## Configuring an Interface as a Private VLAN Host Port

In PVLANS, host ports are part of the secondary VLANs, which are either community VLANs or isolated VLANs. Configuring a PVLAN host port involves two steps. First, you define the port as a PVLAN host port and then you configure a host association between the primary and secondary VLANs.



### Note

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

### Before You Begin

Ensure that the PVLAN feature is enabled.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type* [*chassis/*]*slot/port*
3. switch(config-if)# **switchport mode private-vlan host**
4. switch(config-if)# **switchport private-vlan host-association** {*primary-vlan-id*} {*secondary-vlan-id*}
5. (Optional) switch(config-if)# **no switchport private-vlan host-association**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface</b> <i>type</i> [ <i>chassis/</i> ] <i>slot/port</i>	Selects the port to configure as a PVLAN host port. This port can be on a FEX (identified by the chassis option).
<b>Step 3</b>	switch(config-if)# <b>switchport mode private-vlan host</b>	Configures the port as a host port for a PVLAN.
<b>Step 4</b>	switch(config-if)# <b>switchport private-vlan host-association</b> { <i>primary-vlan-id</i> } { <i>secondary-vlan-id</i> }	Associates the port with the primary and secondary VLANs of a PVLAN. The secondary VLAN can be either an isolated or community VLAN.
<b>Step 5</b>	switch(config-if)# <b>no switchport private-vlan host-association</b>	(Optional) Removes the PVLAN association from the port.

This example shows how to configure Ethernet port 1/12 as a host port for a PVLAN and associate it to primary VLAN 5 and secondary VLAN 101:

```
switch# configure terminal
switch(config)# interface ethernet 1/12
switch(config-if)# switchport mode private-vlan host
switch(config-if)# switchport private-vlan host-association 5 101
```

## Configuring an Interface as a Private VLAN Promiscuous Port

In a PVLAN domain, promiscuous ports are part of the primary VLAN. Configuring a promiscuous port involves two steps. First, you define the port as a promiscuous port and then you configure the mapping between a secondary VLAN and the primary VLAN.

### Before You Begin

Ensure that the PVLAN feature is enabled.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot/port*
3. switch(config-if)# **switchport mode private-vlan promiscuous**
4. switch(config-if)# **switchport private-vlan mapping** {*primary-vlan-id*} {*secondary-vlan-list* | **add** *secondary-vlan-list* | **remove** *secondary-vlan-list*}
5. (Optional) switch(config-if)# **no switchport private-vlan mapping**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface</b> <i>type slot/port</i>	Selects the port to configure as a PVLAN promiscuous port. A physical interface is required. This port cannot be on a FEX.
<b>Step 3</b>	switch(config-if)# <b>switchport mode private-vlan promiscuous</b>	Configures the port as a promiscuous port for a PVLAN. You can only enable a physical Ethernet port as the promiscuous port.
<b>Step 4</b>	switch(config-if)# <b>switchport private-vlan mapping</b> { <i>primary-vlan-id</i> } { <i>secondary-vlan-list</i>   <b>add</b> <i>secondary-vlan-list</i>   <b>remove</b> <i>secondary-vlan-list</i> }	Configures the 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.
<b>Step 5</b>	switch(config-if)# <b>no switchport private-vlan mapping</b>	(Optional) Clears the mapping from the PVLAN.

This example shows how to configure Ethernet interface 1/4 as a promiscuous port associated with primary VLAN 5 and secondary isolated VLAN 200:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# switchport mode private-vlan promiscuous
switch(config-if)# switchport private-vlan mapping 5 200
```

## Configuring a Promiscuous Trunk Port

In a PVLAN domain, promiscuous trunks are part of the primary VLAN. Promiscuous trunk ports can carry multiple primary VLANs. Multiple secondary VLANs under a given primary VLAN can be mapped to a promiscuous trunk port.

Configuring a promiscuous port involves two steps. First, you define the port as a promiscuous port and then you configure the mapping between a secondary VLAN and the primary VLAN. Multiple primary VLANs can be enabled by configuring multiple mappings.



**Note** The number of mappings on a PVLAN trunk port is limited to 16.

### Before You Begin

Ensure that the PVLAN feature is enabled.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot/port*
3. switch(config-if)# **switchport mode private-vlan trunk promiscuous**
4. switch(config-if)# **switchport private-vlan mapping trunk** *{primary-vlan-id}* *{secondary-vlan-id}*
5. (Optional) switch(config-if)# **no switchport private-vlan mapping trunk** *[primary-vlan-id]*

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface</b> <i>type slot/port</i>	Selects the port to configure as a PVLAN promiscuous trunk port.
<b>Step 3</b>	switch(config-if)# <b>switchport mode private-vlan trunk promiscuous</b>	Configures the port as a promiscuous trunk port for a PVLAN. You can only enable a physical Ethernet port as the promiscuous port. This port cannot be on a FEX.
<b>Step 4</b>	switch(config-if)# <b>switchport private-vlan mapping trunk</b> <i>{primary-vlan-id}</i> <i>{secondary-vlan-id}</i>	Maps the trunk port with the primary and secondary VLANs of a PVLAN. The secondary VLAN can be either an isolated or community VLAN.

	Command or Action	Purpose
Step 5	switch(config-if)# <b>no switchport private-vlan mapping trunk</b> [ <i>primary-vlan-id</i> ]	(Optional) Removes the PVLAN mapping from the port. If the <i>primary-vlan-id</i> is not supplied, all PVLAN mappings are removed from the port.

This example shows how to configure Ethernet interface 1/1 as a promiscuous trunk port for a PVLAN and then map the secondary VLANs to the primary VLAN:

```
switch# configure terminal
switch(config)# interface ethernet 1/1
switch(config-if)# switchport mode private-vlan trunk promiscuous
switch(config-if)# switchport private-vlan mapping trunk 5 100
switch(config-if)# switchport private-vlan mapping trunk 5 200
switch(config-if)# switchport private-vlan mapping trunk 6 300
```

## Configuring an Isolated Trunk Port

In a PVLAN domain, isolated trunks are part of a secondary VLAN. Isolated trunk ports can carry multiple isolated VLANs. Only one isolated VLAN under a given primary VLAN can be associated to an isolated trunk port. Configuring an isolated trunk port involves two steps. First, you define the port as an isolated trunk port and then you configure the association between the isolated and primary VLANs. Multiple isolated VLANs can be enabled by configuring multiple associations.

### Before You Begin

Ensure that the PVLAN feature is enabled.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface type** [*chassis*]/*slot*/*port*
3. switch(config-if)# **switchport mode private-vlan trunk** [**secondary**]
4. switch(config-if)# **switchport private-vlan association trunk** {*primary-vlan-id*} {*secondary-vlan-id*}
5. (Optional) switch(config-if)# **no switchport private-vlan association trunk** [*primary-vlan-id*]

### DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	switch(config)# <b>interface type</b> [ <i>chassis</i> ]/ <i>slot</i> / <i>port</i>	Selects the port to configure as a PVLAN isolated trunk port. This port can be on a FEX (identified by the <i>chassis</i> option).
Step 3	switch(config-if)# <b>switchport mode private-vlan trunk</b> [ <b>secondary</b> ]	Configures the port as a secondary trunk port for a PVLAN. <b>Note</b> The <b>secondary</b> keyword is assumed if it is not present.

	Command or Action	Purpose
<b>Step 4</b>	switch(config-if)# <b>switchport private-vlan association trunk</b> { <i>primary-vlan-id</i> } { <i>secondary-vlan-id</i> }	Associates the isolated trunk port with the primary and secondary VLANs of a PVLAN. The secondary VLAN should be an isolated VLAN. Only one isolated VLAN can be mapped under a given primary VLAN.
<b>Step 5</b>	switch(config-if)# <b>no switchport private-vlan association trunk</b> [ <i>primary-vlan-id</i> ]	(Optional) Removes the PVLAN association from the port. If the <i>primary-vlan-id</i> is not supplied, all PVLAN associations are removed from the port.

This example shows how to configure Ethernet interface 1/1 as an isolated trunk port for a PVLAN and then associate the secondary VLANs to the primary VLAN:

```
switch# configure terminal
switch(config)# interface ethernet 1/1
switch(config-if)# switchport mode private-vlan trunk secondary
switch(config-if)# switchport private-vlan association trunk 5 100
switch(config-if)# switchport private-vlan association trunk 6 200
```

## Configuring Private VLANs on FEX Trunk Ports

To enable a FEX HIF configured as a normal dot1q trunk port, the **system private-vlan fex trunk** command must be enabled to allow this interface to forward both primary and secondary VLAN traffic. FEX trunk ports extend the PVLAN domain to all the hosts connected to it and when configured, globally affects all FEX ports connected to the Cisco Nexus device.



### Note

The FEX interface does not support configurations that include promiscuous ports. Also, the FEX interface does not support connections to devices that have promiscuous ports. When promiscuous functionality is required, the device, such as a Cisco Nexus 1000V, must connect to the base ports of the Cisco Nexus device.



### Caution

You must disable all the FEX isolated trunk ports and isolated host ports before configuring PVLANS on the FEX trunk ports. If the FEX isolated trunk ports and the FEX trunk ports are both enabled, unwanted network traffic might occur.

### Before You Begin

Ensure that the PVLAN feature is enabled.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **system private-vlan fex trunk**
3. (Optional) switch(config)# **copy running-config startup-config**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	switch(config)# <b>system private-vlan fex trunk</b>	Enables PVLANS on FEX trunk ports.
Step 3	switch(config)# <b>copy running-config startup-config</b>	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure a PVLAN over a FEX trunk port:

```
switch# configure terminal
switch(config)# system private-vlan fex trunk
switch(config)# copy running-config startup-config
```

## Configuring the Allowed VLANs for PVLAN Trunking Ports

Isolated trunk and promiscuous trunk ports can carry traffic from regular VLANs along with PVLANS.

### Before You Begin

Ensure that the PVLAN feature is enabled.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface type** [*chassis/*]*slot/port*
3. switch(config-if)# **switchport private-vlan trunk allowed vlan** {*vlan-list* | **all** | **none** [**add** | **except** | **none** | **remove** {*vlan-list*}]}

## DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	switch(config)# <b>interface type</b> [ <i>chassis/</i> ] <i>slot/port</i>	Selects the port to configure as a PVLAN host port. This port can be on a FEX (identified by the chassis option).
Step 3	switch(config-if)# <b>switchport private-vlan trunk allowed vlan</b> { <i>vlan-list</i>   <b>all</b>   <b>none</b> [ <b>add</b>   <b>except</b>   <b>none</b>   <b>remove</b> { <i>vlan-list</i> }]}	Sets the allowed VLANs for the private trunk interface. The default is to allow only mapped/associated VLANs on the PVLAN trunk interface.  <b>Note</b> The primary VLANs do not need to be explicitly added to the allowed VLAN list. They are added automatically once there is a mapping between primary and secondary VLANs.

This example shows how to add VLANs to the list of allowed VLANs on an Ethernet PVLAN trunk port:

```
switch# configure terminal
switch(config)# interface ethernet 1/3
switch(config-if)# switchport private-vlan trunk allowed vlan 15-20
```

## Configuring Native 802.1Q VLANs on Private VLANs

Typically, you configure 802.1Q trunks with a native VLAN ID, which strips tagging from all packets on that VLAN. This configuration allows untagged traffic and control traffic to transit the Cisco Nexus device. Secondary VLANs cannot be configured with a native VLAN ID on promiscuous trunk ports. Primary VLANs cannot be configured with a native VLAN ID on isolated trunk ports.



### Note

A trunk can carry the traffic of multiple VLANs. Traffic that belongs to the native VLAN is not encapsulated to transit the trunk. Traffic for other VLANs is encapsulated with tags that identify the VLAN that the traffic belongs to.

### Before You Begin

Ensure that the PVLAN feature is enabled.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type* [*chassis*]/*slot*/*port*
3. switch(config-if)# **switchport private-vlan trunk native** {*vlan vlan-id*}
4. (Optional) switch(config-if)# **no switchport private-vlan trunk native** {*vlan vlan-id*}

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface</b> <i>type</i> [ <i>chassis</i> ]/ <i>slot</i> / <i>port</i>	Selects the port to configure as a PVLAN host port. This port can be on a FEX (identified by the chassis option).
<b>Step 3</b>	switch(config-if)# <b>switchport private-vlan trunk native</b> { <i>vlan vlan-id</i> }	Sets the native VLAN ID for the PVLAN trunk. The default is VLAN 1.
<b>Step 4</b>	switch(config-if)# <b>no switchport private-vlan trunk native</b> { <i>vlan vlan-id</i> }	(Optional) Removes the native VLAN ID from the PVLAN trunk.

## Verifying the Private VLAN Configuration

Use the following commands to display PVLAN configuration information.

Command	Purpose
switch# <b>show feature</b>	Displays the features enabled on the switch.
switch# <b>show interface switchport</b>	Displays information on all interfaces configured as switch ports.
switch# <b>show vlan private-vlan [type]</b>	Displays the status of the PVLAN.

This example shows how to display the PVLAN configuration:

```
switch# show vlan private-vlan
Primary Secondary Type Ports
-----
5 100 community
5 101 community Eth1/12, Eth100/1/1
5 102 community
5 110 community
5 200 isolated Eth1/2
switch# show vlan private-vlan type
Vlan Type
-----
5 primary
100 community
101 community
102 community
110 community
200 isolated
```

This example shows how to display enabled features (some of the output has been removed for brevity):

```
switch# show feature
Feature Name Instance State
-----
fcsp 1 enabled
...
interface-vlan 1 enabled
private-vlan 1 enabled
udld 1 disabled
...
```





# CHAPTER 6

## Configuring Access and Trunk Interfaces

---

This chapter contains the following sections:

- [Information About Access and Trunk Interfaces, page 69](#)
- [Configuring Access and Trunk Interfaces, page 73](#)
- [Verifying the Interface Configuration, page 78](#)

## Information About Access and Trunk Interfaces

### Understanding Access and Trunk Interfaces

Ethernet interfaces can be configured either as access ports or a trunk ports, as follows:

- An access port can have only one VLAN configured on the interface; it can carry traffic for only one VLAN.
- A trunk port can have two or more VLANs configured on the interface; it can carry traffic for several VLANs simultaneously.



**Note**

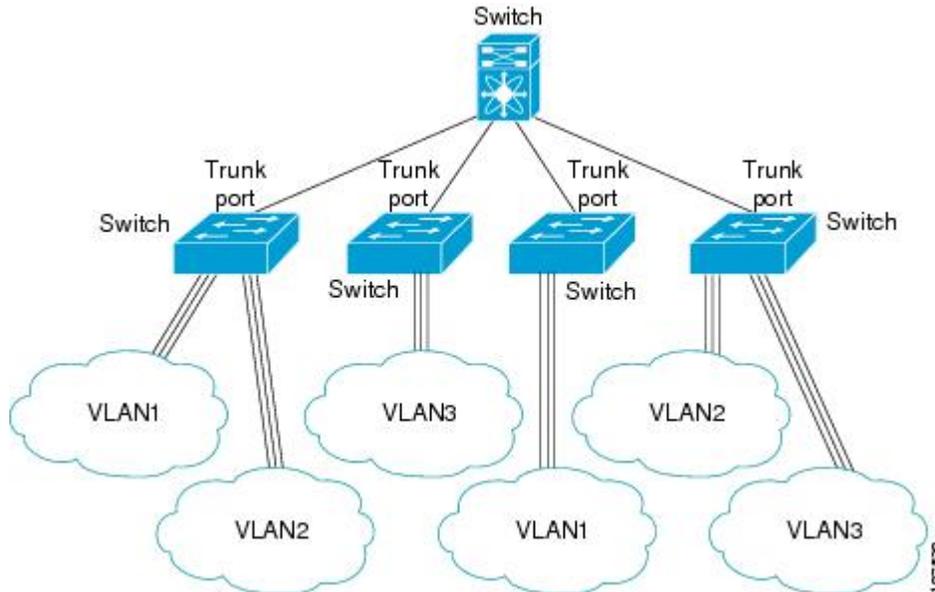
---

Cisco NX-OS supports only IEEE 802.1Q-type VLAN trunk encapsulation.

---

The following figure shows how you can use trunk ports in the network. The trunk port carries traffic for two or more VLANs.

**Figure 5: Devices in a Trunking Environment**



In order to correctly deliver the traffic on a trunk port with several VLANs, the device uses the IEEE 802.1Q encapsulation or tagging method.

To optimize the performance on access ports, you can configure the port as a host port. Once the port is configured as a host port, it is automatically set as an access port, and channel grouping is disabled. Use the host designation to decrease the time it takes the designated port to begin to forward packets.



**Note** Only an end station can be set as a host port; you will receive an error message if you attempt to configure other ports as hosts.

If an access port receives a packet with an 802.1Q tag in the header other than the access VLAN value, that port drops the packet without learning its MAC source address.



**Note** An Ethernet interface can function as either an access port or a trunk port; it cannot function as both port types simultaneously.

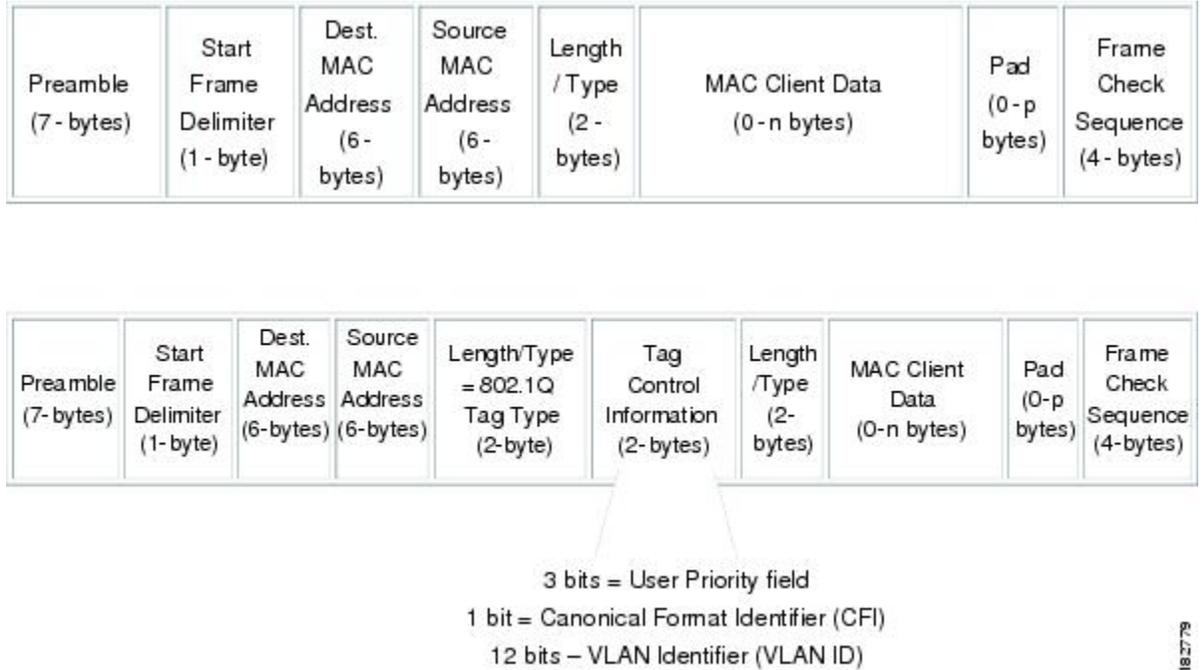
## Understanding IEEE 802.1Q Encapsulation

A trunk is a point-to-point link between the device and another networking device. Trunks carry the traffic of multiple VLANs over a single link and allow you to extend VLANs across an entire network.

To correctly deliver the traffic on a trunk port with several VLANs, the device uses the IEEE 802.1Q encapsulation (tagging) method. This tag carries information about the specific VLAN to which the frame

and packet belong. This method allows packets that are encapsulated for several different VLANs to traverse the same port and maintain traffic separation between the VLANs. The encapsulated VLAN tag also allows the trunk to move traffic end-to-end through the network on the same VLAN.

Figure 6: Header Without and With 802.1Q Tag Included



## Understanding Access VLANs

When you configure a port in access mode, you can specify which VLAN will carry the traffic for that interface. If you do not configure the VLAN for a port in access mode, or an access port, the interface carries traffic for the default VLAN (VLAN1).

You can change the access port membership in a VLAN by specifying the new VLAN. You must create the VLAN before you can assign it as an access VLAN for an access port. If you change the access VLAN on an access port to a VLAN that is not yet created, the system will shut that access port down.



**Note** If you change the VLAN on an access port or a trunk port it will flap the interface. However, if the port is part of a vPC, then first change the native VLAN on the secondary vPC, and then to primary vPC.

If an access port receives a packet with an 802.1Q tag in the header other than the access VLAN value, that port drops the packet without learning its MAC source address.



**Note** If you assign an access VLAN that is also a primary VLAN for a private VLAN, all access ports with that access VLAN will also receive all the broadcast traffic for the primary VLAN in the private VLAN mode.

## Understanding the Native VLAN ID for Trunk Ports

A trunk port can carry untagged packets simultaneously with the 802.1Q tagged packets. When you assign a default port VLAN ID to the trunk port, all untagged traffic travels on the default port VLAN ID for the trunk port, and all untagged traffic is assumed to belong to this VLAN. This VLAN is referred to as the native VLAN ID for a trunk port. The native VLAN ID is the VLAN that carries untagged traffic on trunk ports.

The trunk port sends an egressing packet with a VLAN that is equal to the default port VLAN ID as untagged; all the other egressing packets are tagged by the trunk port. If you do not configure a native VLAN ID, the trunk port uses the default VLAN.

**Note**

---

Native VLAN ID numbers *must* match on both ends of the trunk.

---

## Understanding Allowed VLANs

By default, a trunk port sends traffic to and receives traffic from all VLANs. All VLAN IDs are allowed on each trunk. However, you can remove VLANs from this inclusive list to prevent traffic from the specified VLANs from passing over the trunk. You can add any specific VLANs later that you may want the trunk to carry traffic for back to the list.

To partition the Spanning Tree Protocol (STP) topology for the default VLAN, you can remove VLAN1 from the list of allowed VLANs. Otherwise, VLAN1, which is enabled on all ports by default, will have a very big STP topology, which can result in problems during STP convergence. When you remove VLAN1, all data traffic for VLAN1 on this port is blocked, but the control traffic continues to move on the port.

## Understanding Native 802.1Q VLANs

To provide additional security for traffic passing through an 802.1Q trunk port, the **vlan dot1q tag native** command was introduced. This feature provides a means to ensure that all packets going out of a 802.1Q trunk port are tagged and to prevent reception of untagged packets on the 802.1Q trunk port.

Without this feature, all tagged ingress frames received on a 802.1Q trunk port are accepted as long as they fall inside the allowed VLAN list and their tags are preserved. Untagged frames are tagged with the native VLAN ID of the trunk port before further processing. Only those egress frames whose VLAN tags are inside the allowed range for that 802.1Q trunk port are received. If the VLAN tag on a frame happens to match that of the native VLAN on the trunk port, the tag is stripped off and the frame is sent untagged.

This behavior could potentially be exploited to introduce "VLAN hopping" in which a hacker could try and have a frame jump to a different VLAN. It is also possible for traffic to become part of the native VLAN by sending untagged packets into an 802.1Q trunk port.

To address the above issues, the **vlan dot1q tag native** command performs the following functions:

- On the ingress side, all untagged data traffic is dropped.
- On the egress side, all traffic is tagged. If traffic belongs to native VLAN it is tagged with the native VLAN ID.

This feature is supported on all the directly connected Ethernet and Port Channel interfaces. It is also supported on all the host interface ports of any attached Fabric Extender (FEX).



**Note** You can enable the `vlan dot1q tag native` command by entering the command in the global configuration mode.

## Configuring Access and Trunk Interfaces

### Configuring a LAN Interface as an Ethernet Access Port

You can configure an Ethernet interface as an access port. An access port transmits packets on only one, untagged VLAN. You specify which VLAN traffic that the interface carries. If you do not specify a VLAN for an access port, the interface carries traffic only on the default VLAN. The default VLAN is VLAN1.

The VLAN must exist before you can specify that VLAN as an access VLAN. The system shuts down an access port that is assigned to an access VLAN that does not exist.

#### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *{{type slot/port} | {port-channel number}}*
3. switch(config-if)# **switchport mode** *{access | trunk}*
4. switch(config-if)# **switchport access vlan** *vlan-id*

#### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface</b> <i>{{type slot/port}   {port-channel number}}</i>	Specifies an interface to configure, and enters interface configuration mode.
<b>Step 3</b>	switch(config-if)# <b>switchport mode</b> <i>{access   trunk}</i>	Sets the interface as a nontrunking nontagged single-VLAN Ethernet interface. An access port can carry traffic in one VLAN only. By default, an access port carries traffic for VLAN1; to set the access port to carry traffic for a different VLAN, use the <b>switchport access vlan</b> command.
<b>Step 4</b>	switch(config-if)# <b>switchport access vlan</b> <i>vlan-id</i>	Specifies the VLAN for which this access port will carry traffic. If you do not enter this command, the access port carries traffic on VLAN1 only; use this command to change the VLAN for which the access port carries traffic.

This example shows how to set an interface as an Ethernet access port that carries traffic for a specific VLAN only:

```
switch# configure terminal
switch(config)# interface ethernet 1/10
switch(config-if)# switchport mode access
switch(config-if)# switchport access vlan 5
```

## Configuring Access Host Ports

By using a switchport host, you can make an access port a spanning-tree edge port, and enable BPDU Filtering and BPDU Guard at the same time.

### Before You Begin

Ensure that you are configuring the correct interface; it must be an interface that is connected to an end station.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot/port*
3. switch(config-if)# **switchport host**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface</b> <i>type slot/port</i>	Specifies an interface to configure, and enters interface configuration mode.
<b>Step 3</b>	switch(config-if)# <b>switchport host</b>	Sets the interface to spanning-tree port type edge, turns on BPDU Filtering and BPDU Guard.  <b>Note</b> Apply this command only to switchports that connect to hosts.

This example shows how to set an interface as an Ethernet access host port with EtherChannel disabled:

```
switch# configure terminal
switch(config)# interface ethernet 1/10
switch(config-if)# switchport host
```

## Configuring Trunk Ports

You can configure an Ethernet port as a trunk port; a trunk port transmits untagged packets for the native VLAN plus encapsulated, tagged, packets for multiple VLANs.



**Note** Cisco NX-OS supports only 802.1Q encapsulation.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *{type slot/port | port-channel number}*
3. switch(config-if)# **switchport mode** *{access | trunk}*

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface</b> <i>{type slot/port   port-channel number}</i>	Specifies an interface to configure, and enters interface configuration mode.
<b>Step 3</b>	switch(config-if)# <b>switchport mode</b> <i>{access   trunk}</i>	Sets the interface as an Ethernet trunk port. A trunk port can carry traffic in one or more VLANs on the same physical link (VLANs are based on the trunk-allowed VLANs list). By default, a trunk interface can carry traffic for all VLANs. To specify that only certain VLANs are allowed on the specified trunk, use the <b>switchport trunk allowed vlan</b> command.

This example shows how to set an interface as an Ethernet trunk port:

```
switch# configure terminal
switch(config)# interface ethernet 1/3
switch(config-if)# switchport mode trunk
```

## Configuring the Native VLAN for 802.1Q Trunking Ports

If you do not configure this parameter, the trunk port uses the default VLAN as the native VLAN ID.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *{type slot/port | port-channel number}*
3. switch(config-if)# **switchport trunk native vlan** *vlan-id*

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.

	Command or Action	Purpose
<b>Step 2</b>	switch(config)# <b>interface</b> { <i>type slot/port</i>   <b>port-channel</b> <i>number</i> }	Specifies an interface to configure, and enters interface configuration mode.
<b>Step 3</b>	switch(config-if)# <b>switchport trunk native vlan</b> <i>vlan-id</i>	Sets the native VLAN for the 802.1Q trunk. Valid values are from 1 to 4094, except those VLANs reserved for internal use. The default value is VLAN1.

This example shows how to set the native VLAN for an Ethernet trunk port:

```
switch# configure terminal
switch(config)# interface ethernet 1/3
switch(config-if)# switchport trunk native vlan 5
```

## Configuring the Allowed VLANs for Trunking Ports

You can specify the IDs for the VLANs that are allowed on the specific trunk port.

Before you configure the allowed VLANs for the specified trunk ports, ensure that you are configuring the correct interfaces and that the interfaces are trunks.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** {*type slot/port* | **port-channel** *number*}
3. switch(config-if)# **switchport trunk allowed vlan** {*vlan-list all* | **none** [**add** |**except** | **none** | **remove** {*vlan-list*}]}

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface</b> { <i>type slot/port</i>   <b>port-channel</b> <i>number</i> }	Specifies an interface to configure, and enters interface configuration mode.
<b>Step 3</b>	switch(config-if)# <b>switchport trunk allowed vlan</b> { <i>vlan-list all</i>   <b>none</b> [ <b>add</b>   <b>except</b>   <b>none</b>   <b>remove</b> { <i>vlan-list</i> }]}	<p>Sets allowed VLANs for the trunk interface. The default is to allow all VLANs on the trunk interface: 1 to 3967 and 4048 to 4094. VLANs 3968 to 4047 are the default VLANs reserved for internal use by default; this group of VLANs is configurable. By default, all VLANs are allowed on all trunk interfaces.</p> <p><b>Note</b> You cannot add internally allocated VLANs as allowed VLANs on trunk ports. The system returns a message if you attempt to list an internally allocated VLAN as an allowed VLAN.</p>

This example shows how to add VLANs to the list of allowed VLANs on an Ethernet trunk port:

```
switch# configure terminal
switch(config)# interface ethernet 1/3
switch(config-if)# switchport trunk allow vlan 15-20
```

## Configuring Native 802.1Q VLANs

Typically, you configure 802.1Q trunks with a native VLAN ID, which strips tagging from all packets on that VLAN. This configuration allows all untagged traffic and control traffic to transit the Cisco Nexus device. Packets that enter the switch with 802.1Q tags that match the native VLAN ID value are similarly stripped of tagging.

To maintain the tagging on the native VLAN and drop untagged traffic, enter the **vlan dot1q tag native** command. The switch will tag the traffic received on the native VLAN and admit only 802.1Q-tagged frames, dropping any untagged traffic, including untagged traffic in the native VLAN.

Control traffic continues to be accepted untagged on the native VLAN on a trunked port, even when the **vlan dot1q tag native** command is enabled.



### Note

The **vlan dot1q tag native** command is enabled on global basis.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vlan dot1q tag native [tx-only]**
3. (Optional) switch(config)# **no vlan dot1q tag native [tx-only]**
4. (Optional) switch# **show vlan dot1q tag native**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>vlan dot1q tag native [tx-only]</b>	Enables dot1q (IEEE 802.1Q) tagging for all native VLANs on all trunked ports on the Cisco Nexus device. By default, this feature is disabled.
<b>Step 3</b>	switch(config)# <b>no vlan dot1q tag native [tx-only]</b>	(Optional) Disables dot1q (IEEE 802.1Q) tagging for all native VLANs on all trunked ports on the switch.
<b>Step 4</b>	switch# <b>show vlan dot1q tag native</b>	(Optional) Displays the status of tagging on the native VLANs.

This example shows how to enable 802.1Q tagging on the switch:

```
switch# configure terminal
switch(config)# vlan dot1q tag native
switch(config)# exit
switch# show vlan dot1q tag native
vlan dot1q native tag is enabled
```

## Verifying the Interface Configuration

Use the following commands to display access and trunk interface configuration information.

Command	Purpose
switch# <b>show interface</b>	Displays the interface configuration
switch# <b>show interface switchport</b>	Displays information for all Ethernet interfaces, including access and trunk interfaces.
switch# <b>show interface brief</b>	Displays interface configuration information.



# Configuring Port Channels

This chapter contains the following sections:

- [Information About Port Channels, page 79](#)
- [Configuring Port Channels, page 88](#)
- [Verifying Port Channel Configuration, page 98](#)
- [Verifying the Load-Balancing Outgoing Port ID , page 99](#)

## Information About Port Channels

A port channel bundles individual interfaces into a group to provide increased bandwidth and redundancy. Port channeling also load balances traffic across these physical interfaces. The port channel stays operational as long as at least one physical interface within the port channel is operational.

You create a port channel by bundling compatible interfaces. You can configure and run either static port channels or port channels running the Link Aggregation Control Protocol (LACP).

Any configuration changes that you apply to the port channel are applied to each member interface of that port channel. For example, if you configure Spanning Tree Protocol (STP) parameters on the port channel, Cisco NX-OS applies those parameters to each interface in the port channel.

You can use static port channels, with no associated protocol, for a simplified configuration. For more efficient use of the port channel, you can use the Link Aggregation Control Protocol (LACP), which is defined in IEEE 802.3ad. When you use LACP, the link passes protocol packets.

### Related Topics

[LACP Overview, on page 85](#)

## Understanding Port Channels

Using port channels, Cisco NX-OS provides wider bandwidth, redundancy, and load balancing across the channels.

You can collect ports into a static port channel or you can enable the Link Aggregation Control Protocol (LACP). Configuring port channels with LACP requires slightly different steps than configuring static port

channels. For information on port channel configuration limits, see the *Verified Scalability* document for your platform. For more information about load balancing, see [Load Balancing Using Port Channels](#), on page 82.




---

**Note** Cisco NX-OS does not support Port Aggregation Protocol (PAgP) for port channels.

---

A port channel bundles individual links into a channel group to create a single logical link that provides the aggregate bandwidth of several physical links. If a member port within a port channel fails, traffic previously carried over the failed link switches to the remaining member ports within the port channel.

Each port can be in only one port channel. All the ports in a port channel must be compatible; they must use the same speed and operate in full-duplex mode. When you are running static port channels without LACP, the individual links are all in the on channel mode; you cannot change this mode without enabling LACP.




---

**Note** You cannot change the mode from ON to Active or from ON to Passive.

---

You can create a port channel directly by creating the port-channel interface, or you can create a channel group that acts to aggregate individual ports into a bundle. When you associate an interface with a channel group, Cisco NX-OS creates a matching port channel automatically if the port channel does not already exist. You can also create the port channel first. In this instance, Cisco NX-OS creates an empty channel group with the same channel number as the port channel and takes the default configuration.




---

**Note** A port channel is operationally up when at least one of the member ports is up and that port's status is channeling. The port channel is operationally down when all member ports are operationally down.

---

## Guidelines and Limitations for Port Channel Configuration

Port channels can be configured in one of two ways: either in global configuration mode or in switch profile mode. Consider the following guidelines and limitations when configuring port channels via the configuration synchronization feature in Cisco NX-OS:

- Once a port channel is configured using switch profile mode, it cannot be configured using global configuration (config terminal) mode.




---

**Note** Several port channel sub-commands are not configurable in switch profile mode. These commands can be configured from global configuration mode even if the port channel is created and configured in switch profile mode.

For example, the following command can only be configured in global configuration mode:

```
switchport private-vlan association trunk primary-vlan secondary-vlan
```

---

- Shutdown and no shutdown can be configured in either global configuration mode or switch profile mode.

- If a port channel is created in global configuration mode, channel groups including member interfaces must also be created using global configuration mode.
- Port channels that are configured within switch profile mode may have members both inside and outside of a switch profile.
- If you want to import a member interface to a switch profile, the port channel that corresponds with the member interface must also be present within the switch profile.

For more information on switch profiles, see the .

## Compatibility Requirements

When you add an interface to a port channel group, Cisco NX-OS checks certain interface attributes to ensure that the interface is compatible with the channel group. Cisco NX-OS also checks a number of operational attributes for an interface before allowing that interface to participate in the port-channel aggregation.

The compatibility check includes the following operational attributes:

- Port mode
- Access VLAN
- Trunk native VLAN
- Allowed VLAN list
- Speed
- 802.3x flow control setting
- MTU

The Cisco Nexus device only supports system level MTU. This attribute cannot be changed on an individual port basis.

- Broadcast/Unicast/Multicast Storm Control setting
- Priority-Flow-Control
- Untagged CoS

Use the **show port-channel compatibility-parameters** command to see the full list of compatibility checks that Cisco NX-OS uses.

You can only add interfaces configured with the channel mode set to on to static port channels. You can also only add interfaces configured with the channel mode as active or passive to port channels that are running LACP. You can configure these attributes on an individual member port.

When the interface joins a port channel, the following individual parameters are replaced with the values on the port channel:

- Bandwidth
- MAC address
- Spanning Tree Protocol

The following interface parameters remain unaffected when the interface joins a port channel:

- Description
- CDP
- LACP port priority
- Debounce

After you enable forcing a port to be added to a channel group by entering the **channel-group force** command, the following two conditions occur:

- When an interface joins a port channel, the following parameters are removed and they are operationally replaced with the values on the port channel; however, this change will not be reflected in the running configuration for the interface:
  - QoS
  - Bandwidth
  - Delay
  - STP
  - Service policy
  - ACLs
- When an interface joins or leaves a port channel, the following parameters remain unaffected:
  - Beacon
  - Description
  - CDP
  - LACP port priority
  - Debounce
  - UDLD
  - Shutdown
  - SNMP traps

## Load Balancing Using Port Channels

Cisco NX-OS load balances traffic across all operational interfaces in a port channel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel. Port channels provide load balancing by default.

The basic configuration uses the following criteria to select the link:

- For a Layer 2 frame, it uses the source and destination MAC addresses.
- For a Layer 3 frame, it uses the source and destination MAC addresses and the source and destination IP addresses.

- For a Layer 4 frame, it uses the source and destination MAC addresses and the source and destination IP addresses.



**Note** You have the option to include the source and destination port number for the Layer 4 frame.

You can configure the switch to use one of the following methods (see the following table for more details) to load balance across the port channel:

- Destination MAC address
- Source MAC address
- Source and destination MAC address
- Destination IP address
- Source IP address
- Source and destination IP address
- Destination TCP/UDP port number
- Source TCP/UDP port number
- Source and destination TCP/UDP port number

**Table 5: Port Channel Load-Balancing Criteria**

Configuration	Layer 2 Criteria	Layer 3 Criteria	Layer 4 Criteria
Destination MAC	Destination MAC	Destination MAC	Destination MAC
Source MAC	Source MAC	Source MAC	Source MAC
Source and destination MAC	Source and destination MAC	Source and destination MAC	Source and destination MAC
Destination IP	Destination MAC	Destination MAC, destination IP	Destination MAC, destination IP
Source IP	Source MAC	Source MAC, source IP	Source MAC, source IP
Source and destination IP	Source and destination MAC	Source and destination MAC, source and destination IP	Source and destination MAC, source and destination IP
Destination TCP/UDP port	Destination MAC	Destination MAC, destination IP	Destination MAC, destination IP, destination port
Source TCP/UDP port	Source MAC	Source MAC, source IP	Source MAC, source IP, source port

Configuration	Layer 2 Criteria	Layer 3 Criteria	Layer 4 Criteria
Source and destination TCP/UDP port	Source and destination MAC	Source and destination MAC, source and destination IP	Source and destination MAC, source and destination IP, source and destination port

Fabric Extenders are not configurable individually. Fabric extender configurations are defined on the Cisco Nexus device. In the case of the port-channel load balancing protocol, the table below illustrates which port-channel load balancing option is automatically configured on the fabric extender modules as a result of the configuration performed on the Cisco Nexus device.

The following table shows the criteria used for each configuration:

**Table 6: Port channel Load-Balancing Criteria for the Cisco Nexus 2232 and Cisco Nexus 2248 Fabric Extenders**

Configuration	Layer 2 Criteria	Layer 3 Criteria	Layer 4 Criteria
Destination MAC	Source and destination MAC	Source and destination MAC	Source and destination MAC
Source MAC	Source and destination MAC	Source and destination MAC	Source and destination MAC
Source and destination MAC	Source and destination MAC	Source and destination MAC	Source and destination MAC
Destination IP	Source and destination MAC	Source and destination MAC, and source and destination IP	Source and destination MAC, and source and destination IP
Source IP	Source and destination MAC	Source and destination MAC, and source and destination IP	Source and destination MAC, and source and destination IP
Source and destination IP	Source and destination MAC	Source and destination MAC, and source and destination IP	Source and destination MAC, and source and destination IP
Destination TCP/UDP port	Source and destination MAC	Source and destination MAC, and source and destination IP	Source and destination MAC, source and destination IP, and source and destination port
Source TCP/UDP port	Source and destination MAC	Source and destination MAC, and source and destination IP	Source and destination MAC, source and destination IP, and source and destination port

Configuration	Layer 2 Criteria	Layer 3 Criteria	Layer 4 Criteria
Source and destination TCP/UDP port	Source and destination MAC	Source and destination MAC, source and destination IP	Source and destination MAC, source and destination IP, and source and destination port

Use the option that provides the balance criteria with the greatest variety in your configuration. For example, if the traffic on a port channel is going only to a single MAC address and you use the destination MAC address as the basis of port-channel load balancing, the port channel always chooses the same link in that port channel; using source addresses or IP addresses might result in better load balancing.



**Note** The hardware multicast hw-hash command is not supported on Cisco Nexus 3500 Series switches. It is recommended not to configure this command on these switches.

## Understanding LACP

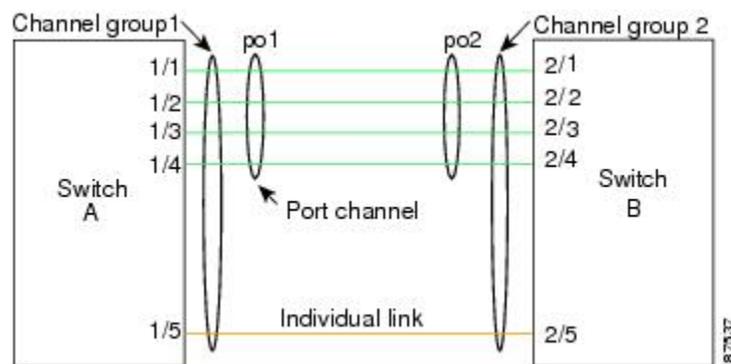
### LACP Overview



**Note** You must enable the LACP feature before you can configure and use LACP functions.

The following figure shows how individual links can be combined into LACP port channels and channel groups as well as function as individual links.

**Figure 7: Individual Links Combined into a Port Channel**



With LACP, just like with static port channels, you can bundle up to 16 interfaces in a channel group.

**Note**


---

When you delete the port channel, Cisco NX-OS automatically deletes the associated channel group. All member interfaces revert to their previous configuration.

---

You cannot disable LACP while any LACP configurations are present.

## LACP ID Parameters

LACP uses the following parameters:

- LACP system priority—Each system that runs LACP has an LACP system priority value. You can accept the default value of 32768 for this parameter, or you can configure a value between 1 and 65535. LACP uses the system priority with the MAC address to form the system ID and also uses the system priority during negotiation with other devices. A higher system priority value means a lower priority.

**Note**


---

The LACP system ID is the combination of the LACP system priority value and the MAC address.

---

- LACP port priority—Each port configured to use LACP has an LACP port priority. You can accept the default value of 32768 for the LACP port priority, or you can configure a value between 1 and 65535. LACP uses the port priority with the port number to form the port identifier. LACP uses the port priority to decide which ports should be put in standby mode when there is a limitation that prevents all compatible ports from aggregating and which ports should be put into active mode. A higher port priority value means a lower priority for LACP. You can configure the port priority so that specified ports have a lower priority for LACP and are most likely to be chosen as active links, rather than hot-standby links.
- LACP administrative key—LACP automatically configures an administrative key value equal to the channel-group number on each port configured to use LACP. The administrative key defines the ability of a port to aggregate with other ports. A port's ability to aggregate with other ports is determined by these factors:
  - Port physical characteristics, such as the data rate, the duplex capability, and the point-to-point or shared medium state
  - Configuration restrictions that you establish

## Channel Modes

Individual interfaces in port channels are configured with channel modes. When you run static port channels, with no protocol, the channel mode is always set to on. After you enable LACP globally on the device, you enable LACP for each channel by setting the channel mode for each interface to active or passive. You can configure either channel mode for individual links in the LACP channel group.

**Note**


---

You must enable LACP globally before you can configure an interface in either the active or passive channel mode.

---

The following table describes the channel modes.

**Table 7: Channel Modes for Individual Links in a Port Channel**

Channel Mode	Description
passive	LACP mode that places a port into a passive negotiating state, in which the port responds to LACP packets that it receives but does not initiate LACP negotiation.
active	LACP mode that places a port into an active negotiating state, in which the port initiates negotiations with other ports by sending LACP packets.
on	All static port channels, that is, that are not running LACP, remain in this mode. If you attempt to change the channel mode to active or passive before enabling LACP, the device returns an error message.  You enable LACP on each channel by configuring the interface in that channel for the channel mode as either active or passive. When an LACP attempts to negotiate with an interface in the on state, it does not receive any LACP packets and becomes an individual link with that interface; it does not join the LACP channel group.

Both the passive and active modes allow LACP to negotiate between ports to determine if they can form a port channel, based on criteria such as the port speed and the trunking state. The passive mode is useful when you do not know whether the remote system, or partner, supports LACP.

Ports can form an LACP port channel when they are in different LACP modes as long as the modes are compatible as in the following examples:

- A port in active mode can form a port channel successfully with another port that is in active mode.
- A port in active mode can form a port channel with another port in passive mode.
- A port in passive mode cannot form a port channel with another port that is also in passive mode because neither port will initiate negotiation.
- A port in on mode is not running LACP.

## LACP Marker Responders

Using port channels, data traffic may be dynamically redistributed due to either a link failure or load balancing. LACP uses the Marker Protocol to ensure that frames are not duplicated or reordered because of this redistribution. Cisco NX-OS supports only Marker Responders.

## LACP-Enabled and Static Port Channel Differences

The following table provides a brief summary of major differences between port channels with LACP enabled and static port channels. For information about the maximum configuration limits, see the *Verified Scalability* document for your device.

**Table 8: Port Channels with LACP Enabled and Static Port Channels**

Configurations	Port Channels with LACP Enabled	Static Port Channels
Protocol applied	Enable globally.	Not applicable.
Channel mode of links	Can be either: <ul style="list-style-type: none"> <li>• Active</li> <li>• Passive</li> </ul>	Can only be On.

# Configuring Port Channels

## Creating a Port Channel

You can create a port channel before creating a channel group. Cisco NX-OS automatically creates the associated channel group.



**Note**

If you want LACP-based port channels, you need to enable LACP.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface port-channel** *channel-number*
3. switch(config)# **no interface port-channel** *channel-number*

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface port-channel</b> <i>channel-number</i>	Specifies the port-channel interface to configure, and enters the interface configuration mode. The range is from 1 to 4096. Cisco NX-OS automatically creates the channel group if it does not already exist.

	Command or Action	Purpose
<b>Step 3</b>	switch(config)# <b>no interface port-channel</b> <i>channel-number</i>	Removes the port channel and deletes the associated channel group.

This example shows how to create a port channel:

```
switch# configure terminal
switch (config)# interface port-channel 1
```

## Adding a Port to a Port Channel

You can add a port to a new channel group or to a channel group that already contains ports. Cisco NX-OS creates the port channel associated with this channel group if the port channel does not already exist.



**Note** If you want LACP-based port channels, you need to enable LACP.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot/port*
3. (Optional) switch(config-if)# **switchport mode trunk**
4. (Optional) switch(config-if)# **switchport trunk** {**allowed vlan** *vlan-id* | **native vlan** *vlan-id*}
5. switch(config-if)# **channel-group** *channel-number*
6. (Optional) switch(config-if)# **no channel-group**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface</b> <i>type slot/port</i>	Specifies the interface that you want to add to a channel group and enters the interface configuration mode.
<b>Step 3</b>	switch(config-if)# <b>switchport mode trunk</b>	(Optional) Configures the interface as a trunk port.
<b>Step 4</b>	switch(config-if)# <b>switchport trunk</b> { <b>allowed vlan</b> <i>vlan-id</i>   <b>native vlan</b> <i>vlan-id</i> }	(Optional) Configures necessary parameters for a trunk port.
<b>Step 5</b>	switch(config-if)# <b>channel-group</b> <i>channel-number</i>	Configures the port in a channel group and sets the mode. The channel-number range is from 1 to 4096. Cisco NX-OS creates the port channel associated with this channel group if the port channel does not already exist. This is called implicit port channel creation.

	Command or Action	Purpose
<b>Step 6</b>	switch(config-if)# <b>no channel-group</b>	(Optional) Removes the port from the channel group. The port reverts to its original configuration.

This example shows how to add an Ethernet interface 1/4 to channel group 1:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# switchport mode trunk
switch(config-if)# channel-group 1
```

## Configuring Load Balancing Using Port Channels

You can configure the load-balancing algorithm for port channels that applies to the entire device.



### Note

If you want LACP-based port channels, you need to enable LACP.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **port-channel load-balance ethernet** {[**destination-ip** | **destination-mac** | **destination-port** | **source-dest-ip** | **source-dest-mac** | **source-dest-port** | **source-ip** | **source-mac** | **source-port**] | **crc-poly**}
3. (Optional) switch(config)# **no port-channel load-balance ethernet**
4. (Optional) switch# **show port-channel load-balance**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>port-channel load-balance ethernet</b> {[ <b>destination-ip</b>   <b>destination-mac</b>   <b>destination-port</b>   <b>source-dest-ip</b>   <b>source-dest-mac</b>   <b>source-dest-port</b>   <b>source-ip</b>   <b>source-mac</b>   <b>source-port</b> ]   <b>crc-poly</b> }	Specifies the load-balancing algorithm for the device. The range depends on the device. The default is <b>source-dest-mac</b> . <b>Note</b>
<b>Step 3</b>	switch(config)# <b>no port-channel load-balance ethernet</b>	(Optional) Restores the default load-balancing algorithm of source-dest-mac.
<b>Step 4</b>	switch# <b>show port-channel load-balance</b>	(Optional) Displays the port-channel load-balancing algorithm.

This example shows how to configure source IP load balancing for port channels:

```
switch# configure terminal
switch (config)# port-channel load-balance ethernet source-ip
```

## Configuring Hardware Hashing for Multicast Traffic

By default, ingress multicast traffic on any port in the switch selects a particular port channel member to egress the traffic. You can configure hardware hashing for multicast traffic to reduce potential bandwidth issues and to provide effective load balancing of the ingress multicast traffic. Use the **hardware multicast hw-hash** command to enable hardware hashing. To restore the default, use the **no hardware multicast hw-hash** command.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface port-channel** *channel-number*
3. switch(config-if)# **[no] hardware multicast hw-hash**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface port-channel</b> <i>channel-number</i>	Selects the port channel and enters the interface configuration mode.
<b>Step 3</b>	switch(config-if)# <b>[no] hardware multicast hw-hash</b>	Configures hardware hashing for the specified port channel.

This example shows how to configure hardware hashing on a port channel:

```
switch# configure terminal
switch (config)# interface port-channel 21
switch(config-if)# hardware multicast hw-hash
```

This example shows how to remove hardware hashing from a port channel:

```
switch# configure terminal
switch (config)# interface port-channel 21
switch(config-if)# no hardware multicast hw-hash
```

## Enabling LACP

LACP is disabled by default; you must enable LACP before you begin LACP configuration. You cannot disable LACP while any LACP configuration is present.

LACP learns the capabilities of LAN port groups dynamically and informs the other LAN ports. Once LACP identifies correctly matched Ethernet links, it facilitates grouping the links into an port channel. The port channel is then added to the spanning tree as a single bridge port.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **feature lacp**
3. (Optional) switch(config)# **show feature**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>feature lacp</b>	Enables LACP on the switch.
<b>Step 3</b>	switch(config)# <b>show feature</b>	(Optional) Displays enabled features.

This example shows how to enable LACP:

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

## Configuring the Channel Mode for a Port

You can configure the channel mode for each individual link in the LACP port channel as active or passive. This channel configuration mode allows the link to operate with LACP.

When you configure port channels with no associated protocol, all interfaces on both sides of the link remain in the on channel mode.

### Before You Begin

Ensure that you have enabled the LACP feature.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot/port*
3. switch(config-if)# **channel-group** *channel-number* [**force**] [**mode** {**on** | **active** | **passive**}]
4. switch(config-if)# **no channel-group** *number mode*

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface</b> <i>type slot/port</i>	Specifies the interface to configure, and enters the interface configuration mode.
<b>Step 3</b>	switch(config-if)# <b>channel-group</b> <i>channel-number</i> [ <b>force</b> ] [ <b>mode</b> { <b>on</b>   <b>active</b>   <b>passive</b> }]	<p>Specifies the port mode for the link in a port channel. After LACP is enabled, you configure each link or the entire channel as active or passive.</p> <p><b>force</b>—Specifies that the LAN port be forcefully added to the channel group.</p> <p><b>mode</b>—Specifies the port channel mode of the interface.</p> <p><b>active</b>—Specifies that when you enable LACP, this command enables LACP on the specified interface. The interface is in an active negotiating state in which the port initiates negotiations with other ports by sending LACP packets.</p> <p><b>on</b>—(Default mode) Specifies that all port channels that are not running LACP remain in this mode.</p> <p><b>passive</b>—Enables LACP only if an LACP device is detected. The interface is in a passive negotiation state in which the port responds to LACP packets that it receives but does not initiate LACP negotiation.</p> <p>When you run port channels with no associated protocol, the channel mode is always on.</p>
<b>Step 4</b>	switch(config-if)# <b>no channel-group</b> <i>number</i> <b>mode</b>	Returns the port mode to on for the specified interface.

This example shows how to set the LACP-enabled interface to active port-channel mode for Ethernet interface 1/4 in channel group 5:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# channel-group 5 mode active
```

This example shows how to forcefully add an interface to the channel group 5:

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

## Configuring the LACP Fast Timer Rate

You can change the LACP timer rate to modify the duration of the LACP timeout. Use the **lACP rate** command to set the rate at which LACP control packets are sent to an LACP-supported interface. You can change the timeout rate from the default rate (30 seconds) to the fast rate (1 second). This command is supported only on LACP-enabled interfaces.

### Before You Begin

Ensure that you have enabled the LACP feature.

**SUMMARY STEPS**

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot/port*
3. switch(config-if)# **lacp rate fast**

**DETAILED STEPS**

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface</b> <i>type slot/port</i>	Specifies the interface to configure and enters the interface configuration mode.
<b>Step 3</b>	switch(config-if)# <b>lacp rate fast</b>	Configures the fast rate (one second) at which LACP control packets are sent to an LACP-supported interface.

This example shows how to configure the LACP fast rate on Ethernet interface 1/4:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# lacp rate fast
```

This example shows how to restore the LACP default rate (30 seconds) on Ethernet interface 1/4.

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# no lacp rate fast
```

**Configuring the LACP System Priority and System ID**

The LACP system ID is the combination of the LACP system priority value and the MAC address.

**Before You Begin**

Ensure that you have enabled the LACP feature.

**SUMMARY STEPS**

1. switch# **configure terminal**
2. switch(config)# **lacp system-priority** *priority*
3. (Optional) switch# **show lacp system-identifier**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>lACP system-priority</b> <i>priority</i>	Configures the system priority for use with LACP. Valid values are 1 through 65535, and higher numbers have lower priority. The default value is 32768.
<b>Step 3</b>	switch# <b>show lACP system-identifier</b>	(Optional) Displays the LACP system identifier.

This example shows how to set the LACP system priority to 2500:

```
switch# configure terminal
switch(config)# lACP system-priority 2500
```

## Configuring the LACP Port Priority

You can configure each link in the LACP port channel for the port priority.

### Before You Begin

Ensure that you have enabled the LACP feature.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot/port*
3. switch(config-if)# **lACP port-priority** *priority*

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface</b> <i>type slot/port</i>	Specifies the interface to configure, and enters the interface configuration mode.
<b>Step 3</b>	switch(config-if)# <b>lACP port-priority</b> <i>priority</i>	Configures the port priority for use with LACP. Valid values are 1 through 65535, and higher numbers have lower priority. The default value is 32768.

This example shows how to set the LACP port priority for Ethernet interface 1/4 to 40000:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# lacp port priority 40000
```

## Disabling LACP Graceful Convergence

### Before You Begin

- Enable the LACP feature.
- Confirm that the port channel is in the administratively down state.
- Ensure that you are in the correct VDC. To switch to the correct VDC, enter the **switchto vdc** command.

### SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *number*
3. **shutdown**
4. **no lacp graceful-convergence**
5. **no shutdown**
6. (Optional) **copy running-config startup-config**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>configure terminal</b>  <b>Example:</b> switch# configure terminal switch(config)#	Enters global configuration mode.
<b>Step 2</b>	<b>interface port-channel</b> <i>number</i>  <b>Example:</b> switch(config)# interface port-channel 1 switch(config) #	Specifies the port channel interface to configure, and enters interface configuration mode.
<b>Step 3</b>	<b>shutdown</b>  <b>Example:</b> switch(config-if)# shutdown switch(config-if) #	Administratively shuts down the port channel.
<b>Step 4</b>	<b>no lacp graceful-convergence</b>  <b>Example:</b> switch(config-if)# no lacp graceful-convergence switch(config-if) #	Disables LACP graceful convergence on the specified port channel.

	Command or Action	Purpose
Step 5	<b>no shutdown</b>  <b>Example:</b> switch(config-if) # no shutdown switch(config-if) #	Administratively brings the port channel up.
Step 6	<b>copy running-config startup-config</b>  <b>Example:</b> switch(config-if) # copy running-config startup-config	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

The following example disables LACP graceful convergence on a port channel:

```
switch# configure terminal
switch(config) # interface port-channel 1
switch(config-if) # shutdown
switch(config-if) # no lacp graceful-convergence
switch(config-if) # no shutdown
switch(config-if) #
```

## Reenabling LACP Graceful Convergence

### Before You Begin

- Enable the LACP feature.
- Confirm that the port channel is in the administratively down state.
- Ensure that you are in the correct VDC. To switch to the correct VDC, enter the **switchto vdc** command.

### SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *number*
3. **shutdown**
4. **lacp graceful-convergence**
5. **no shutdown**
6. (Optional) **copy running-config startup-config**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>configure terminal</b>  <b>Example:</b> switch# configure terminal switch(config)#	Enters global configuration mode.
<b>Step 2</b>	<b>interface port-channel <i>number</i></b>  <b>Example:</b> switch(config)# interface port-channel 1 switch(config)#	Specifies the port channel interface to configure, and enters interface configuration mode.
<b>Step 3</b>	<b>shutdown</b>  <b>Example:</b> switch(config-if)# shutdown switch(config-if)#	Administratively shuts down the port channel.
<b>Step 4</b>	<b>lACP graceful-convergence</b>  <b>Example:</b> switch(config-if)# lACP graceful-convergence switch(config-if)#	Enables LACP graceful convergence on the specified port channel.
<b>Step 5</b>	<b>no shutdown</b>  <b>Example:</b> switch(config-if)# no shutdown switch(config-if)#	Administratively brings the port channel up.
<b>Step 6</b>	<b>copy running-config startup-config</b>  <b>Example:</b> switch(config-if)# copy running-config startup-config	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

The following example disables LACP graceful convergence on a port channel:

```
switch# configure terminal
switch(config)# interface port-channel 1
switch(config-if)# shutdown
switch(config-if)# lACP graceful-convergence
switch(config-if)# no shutdown
switch(config-if)#
```

## Verifying Port Channel Configuration

Use the following command to verify the port channel configuration information:

Command	Purpose
<b>show interface port channel</b> <i>channel-number</i>	Displays the status of a port channel interface.
<b>show feature</b>	Displays enabled features.
<b>show resource</b>	Displays the number of resources currently available in the system.
<b>show lacp</b> { <b>counters</b>   <b>interface</b> <i>type slot/port</i>   <b>neighbor</b>   <b>port-channel</b>   <b>system-identifier</b> }	Displays LACP information.
<b>show port-channel compatibility-parameters</b>	Displays the parameters that must be the same among the member ports in order to join a port channel.
<b>show port-channel database</b> [ <b>interface port-channel</b> <i>channel-number</i> ]	Displays the aggregation state for one or more port-channel interfaces.
<b>show port-channel summary</b>	Displays a summary for the port channel interfaces.
<b>show port-channel traffic</b>	Displays the traffic statistics for port channels.
<b>show port-channel usage</b>	Displays the range of used and unused channel numbers.
<b>show port-channel database</b>	Displays information on current running of the port channel feature.
<b>show port-channel load-balance</b>	Displays information about load-balancing using port channels.

## Verifying the Load-Balancing Outgoing Port ID

### Command Guidelines

The **show port-channel load-balance** command allows you to verify which ports a given frame is hashed to on a port channel. You need to specify the VLAN and the destination MAC in order to get accurate results.



#### Note

Certain traffic flows are not subject to hashing such as when there is a single port in a port-channel.

To display the load-balancing outgoing port ID, perform one of the tasks:

Command	Purpose
switch# <b>show port-channel load-balance forwarding-path interface port-channel</b> <i>port-channel-id</i> <b>vlan</b> <i>vlan-id</i> <b>dst-ip</b> <i>src-ip</i> <b>dst-mac</b> <b>src-mac</b> <i>l4-src-port</i> <i>port-id</i> <b>l4-dst-port</b> <i>port-id</i>	Displays the outgoing port ID.

### Example

This example shows how to display the load balancing outgoing port ID:

```
switch# show port-channel load-balance forwarding-path interface port-channel 10 vlan 1
dst-ip 1.225.225.225 src-ip 1.1.10.10 src-mac aa:bb:cc:dd:ee:ff
14-src-port 0 14-dst-port 1
Missing params will be substituted by 0's. Load-balance Algorithm on switch: source-dest-port
  crc8_hash:204 Outgoing port id: Ethernet 1/1 Param(s) used to calculate load balance:
dst-port: 0
src-port: 0
dst-ip: 1.225.225.225
src-ip: 1.1.10.10
dst-mac: 0000.0000.0000
src-mac: aabb.ccdd.eeff
```



## CHAPTER 8

# Configuring Virtual Port Channels

---

This chapter contains the following sections:

- [Information About vPCs, page 101](#)
- [Guidelines and Limitations for vPCs, page 115](#)
- [Configuring vPCs, page 116](#)
- [Configuring the vPC Peer Switch, page 134](#)
- [Verifying the vPC Configuration, page 138](#)
- [vPC Example Configurations, page 143](#)
- [vPC Default Settings, page 147](#)

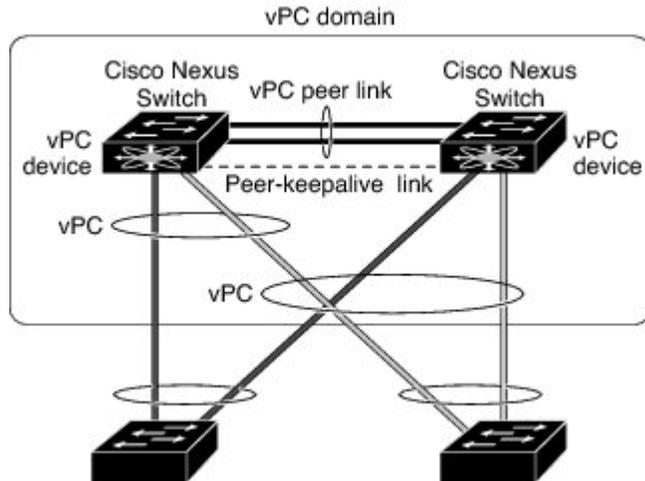
## Information About vPCs

### vPC Overview

A virtual port channel (vPC) allows links that are physically connected to two different Cisco Nexus devices or Cisco Nexus Fabric Extenders to appear as a single port channel by a third device (see the following figure). The third device can be a switch, server, or any other networking device. You can configure vPCs in topologies that include Cisco Nexus devices connected to Cisco Nexus Fabric Extenders. A vPC can provide multipathing,

which allows you to create redundancy by enabling multiple parallel paths between nodes and load balancing traffic where alternative paths exist.

**Figure 8: vPC Architecture**



You configure the EtherChannels by using one of the following:

- No protocol
- Link Aggregation Control Protocol (LACP)

When you configure the EtherChannels in a vPC—including the vPC peer link channel—each switch can have up to 16 active links in a single EtherChannel. When you configure a vPC on a Fabric Extender, only one port is allowed in an EtherChannel.



**Note** You must enable the vPC feature before you can configure or run the vPC functionality.

To enable the vPC functionality, you must create a peer-keepalive link and a peer-link under the vPC domain for the two vPC peer switches to provide the vPC functionality.

To create a vPC peer link you configure an EtherChannel on one Cisco Nexus device by using two or more Ethernet ports. On the other switch, you configure another EtherChannel again using two or more Ethernet ports. Connecting these two EtherChannels together creates a vPC peer link.



**Note** We recommend that you configure the vPC peer-link EtherChannels as trunks.

The vPC domain includes both vPC peer devices, the vPC peer-keepalive link, the vPC peer link, and all of the EtherChannels in the vPC domain connected to the downstream device. You can have only one vPC domain ID on each vPC peer device.



**Note** Always attach all vPC devices using EtherChannels to both vPC peer devices.

A vPC provides the following benefits:

- Allows a single device to use an EtherChannel across two upstream devices
- Eliminates Spanning Tree Protocol (STP) blocked ports
- Provides a loop-free topology
- Uses all available uplink bandwidth
- Provides fast convergence if either the link or a switch fails
- Provides link-level resiliency
- Assures high availability

## Terminology

### vPC Terminology

The terminology used in vPCs is as follows:

- vPC—combined EtherChannel between the vPC peer devices and the downstream device.
- vPC peer device—One of a pair of devices that are connected with the special EtherChannel known as the vPC peer link.
- vPC peer link—link used to synchronize states between the vPC peer devices.
- vPC member port—Interfaces that belong to the vPCs.
- Host vPC port—Fabric Extender host interfaces that belong to a vPC.
- vPC domain—domain that includes both vPC peer devices, the vPC peer-keepalive link, and all of the port channels in the vPC connected to the downstream devices. It is also associated to the configuration mode that you must use to assign vPC global parameters. The vPC domain ID must be the same on both switches.
- vPC peer-keepalive link—The peer-keepalive link monitors the vitality of a vPC peer Cisco Nexus device. The peer-keepalive link sends configurable, periodic keepalive messages between vPC peer devices.

No data or synchronization traffic moves over the vPC peer-keepalive link; the only traffic on this link is a message that indicates that the originating switch is operating and running vPCs.

### Fabric Extender Terminology

The terminology used for the Cisco Nexus Fabric Extender is as follows:

- Fabric interface—A 10-Gigabit Ethernet uplink port designated for connection from the Fabric Extender to its parent switch. A fabric interface cannot be used for any other purpose. It must be directly connected to the parent switch.
- EtherChannel fabric interface—An EtherChannel uplink connection from the Fabric Extender to its parent switch. This connection consists of fabric interfaces bundled into a single logical channel.

- Host interface—An Ethernet interface for server or host connectivity. These ports are 1-Gigabit Ethernet interfaces or 10-Gigabit Ethernet interfaces, depending on the fabric extender model.
- EtherChannel host interface—An EtherChannel downlink connection from the Fabric Extender host interface to a server port.

**Note**

An EtherChannel host interface consists of only one host interface and can be configured either as a Link Aggregation Control Protocol (LACP) or non-LACP EtherChannel.

## Supported vPC Topologies

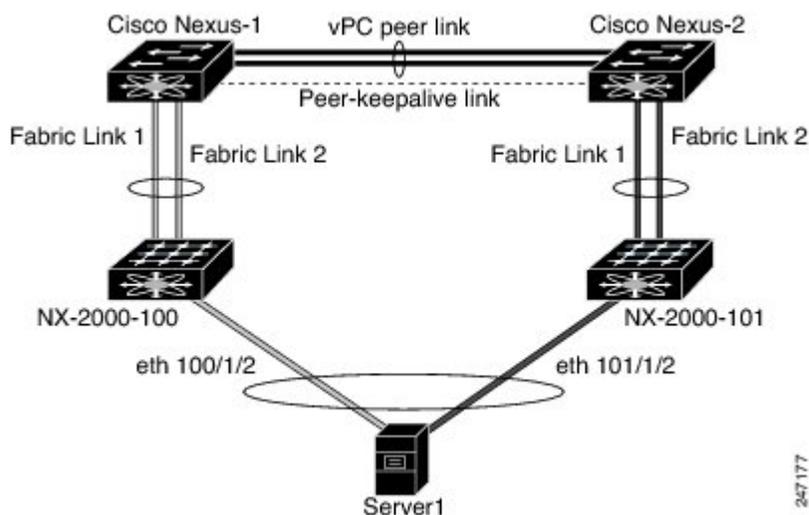
### Cisco Nexus Device vPC Topology

#### Single Homed Fabric Extender vPC Topology

You can connect a server with dual or quad or more network adapters that are configured in a vPC to a pair of Cisco Nexus Fabric Extenders which are connected to the Cisco Nexus devices as depicted. Depending on the FEX model, you may be able to connect one or more network adapter interfaces to each fabric extender. As an example, the following figure refers to a topology built with the Cisco Nexus 2148T fabric extender, where a server has one link only to each fabric extender. A topology with Cisco Nexus 2248TP or with Cisco Nexus 2232PP fabric extender could consist of more links from the server to a single fabric extender.

The topology that is shown in the following figure provides the vPC functionality to dual homed servers with 1-Gigabit Ethernet uplink interfaces.

**Figure 9: Single Homed Fabric Extender vPC Topology**



The Cisco Nexus device can support up to 12 configured single homed Fabric Extenders (576 ports) with this topology however only 480 576 dual homed host servers can be configured in a vPCs with this configuration.

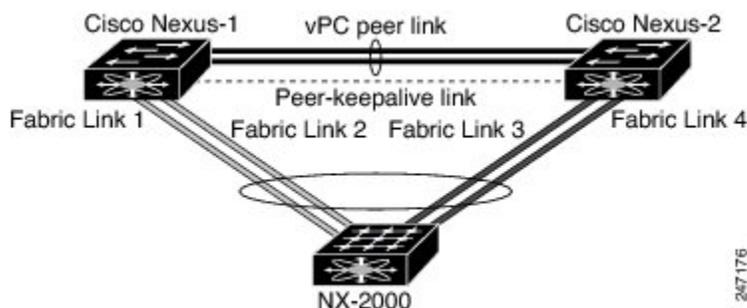
**Note**

The Cisco Nexus 2148T fabric extender does not support EtherChannels on its host interfaces. Therefore a maximum of two links can be configured in an EtherChannel from the server where each link is connected to a separate Fabric Extender.

## Dual Homed Fabric Extender vPC Topology

You can connect the Cisco Nexus Fabric Extender to two upstream Cisco Nexus devices and downstream to a number of single homed servers. The topology shown in the following figure provides the vPC functionality to singly connected servers with 1-Gigabit Ethernet uplink interfaces.

**Figure 10: Dual Homed Fabric Extender vPC Topology**



The Cisco Nexus device can support up to 12 configured dual homed Fabric Extenders with this topology. A maximum of 576 single homed servers can be connected to this configuration.

## vPC Domain

To create a vPC domain, you must first create a vPC domain ID on each vPC peer switch using a number from 1 to 1000. This ID must be the same on a set of vPC peer devices.

You can configure the EtherChannels and vPC peer links by using LACP or no protocol. When possible, we recommend that you use LACP on the peer-link, because LACP provides configuration checks against a configuration mismatch on the EtherChannel.

The vPC peer switches use the vPC domain ID that you configure to automatically assign a unique vPC system MAC address. Each vPC domain has a unique MAC address that is used as a unique identifier for the specific vPC-related operations, although the switches use the vPC system MAC addresses only for link-scope operations, such as LACP. We recommend that you create each vPC domain within the contiguous network with a unique domain ID. You can also configure a specific MAC address for the vPC domain, rather than having the Cisco NX-OS software assign the address.

The vPC peer switches use the vPC domain ID that you configure to automatically assign a unique vPC system MAC address. The switches use the vPC system MAC addresses only for link-scope operations, such as LACP or BPDUs. You can also configure a specific MAC address for the vPC domain.

We recommend that you configure the same VPC domain ID on both peers and, the domain ID should be unique in the network. For example, if there are two different VPCs (one in access and one in aggregation) then each vPC should have a unique domain ID.

After you create a vPC domain, the Cisco NX-OS software automatically creates a system priority for the vPC domain. You can also manually configure a specific system priority for the vPC domain.

**Note**

If you manually configure the system priority, you must ensure that you assign the same priority value on both vPC peer switches. If the vPC peer switches have different system priority values, the vPC will not come up.

## Peer-Keepalive Link and Messages

The Cisco NX-OS software uses a peer-keepalive link between the vPC peers to transmit periodic, configurable keepalive messages. You must have Layer 3 connectivity between the peer switches to transmit these messages; the system cannot bring up the vPC peer link unless a peer-keepalive link is already up and running.

You can configure a hold-timeout and a timeout value simultaneously.

**Hold-timeout value**—The hold-timeout value range is between 3 to 10 seconds, with a default value of 3 seconds. This timer starts when the vPC peer link goes down. The purpose of the hold-timeout period is to prevent false-positive cases.

If you configure a hold-timeout value that is lower than the timeout value, then the vPC system ignores vPC peer-keepalive messages for the hold-timeout period and considers messages for the remainder of the timeout period. If no keepalive message is received for this period, the vPC secondary device takes over the role of the primary device. For example, if the hold-timeout value is 3 seconds and the timeout value is 5 seconds, for the first 3 seconds vPC keepalive messages are ignored (such as, when accommodating a supervisor failure for a few seconds after peer link failure) and keepalive messages are considered for the remaining timeout period of 2 seconds. After this period, the vPC secondary device takes over as the primary device, in case there is no keep alive message.

**Timeout value**—The timeout value range is between 3 to 20 seconds, with a default value of 5 seconds. This timer starts at the end of the hold-timeout interval. If you configure a timeout value that is lower than or equal to the hold-timeout value, then the timeout duration is initiated after the hold-timeout period. For example, if the timeout value is 3 seconds and the hold-timeout value is 5 seconds, the timeout period starts after 5 seconds.

**Note**

We recommend that you configure the vPC peer-keepalive link on the Cisco Nexus device to run in the management VRF using the mgmt 0 interfaces. If you configure the default VRF, ensure that the vPC peer link is not used to carry the vPC peer-keepalive messages.

## Compatibility Parameters for vPC Peer Links

Many configuration and operational parameters must be identical on all interfaces in the vPC. After you enable the vPC feature and configure the peer link on both vPC peer switches, Cisco Fabric Services (CFS) messages provide a copy of the configuration on the local vPC peer switch configuration to the remote vPC peer switch. The system then determines whether any of the crucial configuration parameters differ on the two switches.

Enter the **show vpc consistency-parameters** command to display the configured values on all interfaces in the vPC. The displayed configurations are only those configurations that would limit the vPC peer link and vPC from coming up.

The compatibility check process for vPCs differs from the compatibility check for regular EtherChannels.

## Configuration Parameters That Must Be Identical

The configuration parameters in this section must be configured identically on both switches at either end of the vPC peer link.



### Note

You must ensure that all interfaces in the vPC have the identical operational and configuration parameters listed in this section.

Enter the **show vpc consistency-parameters** command to display the configured values on all interfaces in the vPC. The displayed configurations are only those configurations that would limit the vPC peer link and vPC from coming up.

The switch automatically checks for compatibility of these parameters on the vPC interfaces. The per-interface parameters must be consistent per interface, and the global parameters must be consistent globally.

- Port-channel mode: on, off, or active
- Link speed per channel
- Duplex mode per channel
- Trunk mode per channel:
  - Native VLAN
  - VLANs allowed on trunk
  - Tagging of native VLAN traffic
- Spanning Tree Protocol (STP) mode
- STP region configuration for Multiple Spanning Tree (MST)
- Enable or disable state per VLAN
- STP global settings:
  - Bridge Assurance setting
  - Port type setting—We recommend that you set all vPC interfaces as normal ports
  - Loop Guard settings
- STP interface settings:
  - Port type setting
  - Loop Guard
  - Root Guard
- For the Fabric Extender vPC topology, all the interface level parameters mentioned above should be identically configured for host interface from both the switches.

- Fabric Extender FEX number configured on an EtherChannel fabric interface; for the Fabric Extender vPC topology.

If any of these parameters are not enabled or defined on either switch, the vPC consistency check ignores those parameters.

**Note**

To ensure that none of the vPC interfaces are in the suspend mode, enter the **show vpc brief** and **show vpc consistency-parameters** commands and check the syslog messages.

## Configuration Parameters That Should Be Identical

When any of the following parameters are not configured identically on both vPC peer switches, a misconfiguration might cause undesirable behavior in the traffic flow:

- MAC aging timers
- Static MAC entries
- VLAN interface—Each switch on the end of the vPC peer link must have a VLAN interface configured for the same VLAN on both ends and they must be in the same administrative and operational mode. Those VLANs configured on only one switch of the peer link do not pass traffic using the vPC or peer link. You must create all VLANs on both the primary and secondary vPC switches, or the VLAN will be suspended.
- Private VLAN configuration
- All ACL configurations and parameters
- Quality of service (QoS) configuration and parameters—Local parameters; global parameters must be identical
- STP interface settings:
  - BPDU Filter
  - BPDU Guard
  - Cost
  - Link type
  - Priority
  - VLANs (Rapid PVST+)

To ensure that all the configuration parameters are compatible, we recommend that you display the configurations for each vPC peer switch once you configure the vPC.

## Graceful Type-1 Check

When a consistency check fails, vPCs are brought down only on the secondary vPC switch. The VLANs remain up on the primary switch and Type-1 configurations can be performed without traffic disruption. This feature is used both in the case of global as well as interface-specific Type-1 inconsistencies.

This feature is not enabled for dual-active FEX ports. When a Type-1 mismatch occurs, VLANs are suspended on these ports on both switches.

## Per-VLAN Consistency Check

Beginning with Cisco NX-OS Release 5.0(2)N2(1), some Type-1 consistency checks are performed on a per-VLAN basis when spanning tree is enabled or disabled on a VLAN. VLANs that do not pass the consistency check are brought down on both the primary and secondary switches while other VLANs are not affected.

## vPC Auto-Recovery

Beginning with Cisco NX-OS Release 5.0(2)N2(1), the vPC auto-recovery feature re-enables vPC links in the following scenarios:

When both vPC peer switches reload and only one switch reboots, auto-recovery allows that switch to assume the role of the primary switch and the vPC links will be allowed to come up after a predetermined period of time. The reload delay period in this scenario can range from 240 to 3600 seconds.

When vPCs are disabled on a secondary vPC switch due to a peer-link failure and then the primary vPC switch fails or is unable to forward traffic, the secondary switch reenables the vPCs. In this scenario, the vPC waits for three consecutive keepalive failures to recover the vPC links.

## vPC Peer Links

A vPC peer link is the link that is used to synchronize the states between the vPC peer devices.

**Note**

---

You must configure the peer-keepalive link before you configure the vPC peer link or the peer link will not come up.

---

### vPC Peer Link Overview

You can have only two switches as vPC peers; each switch can serve as a vPC peer to only one other vPC peer. The vPC peer switches can also have non-vPC links to other switches.

To make a valid configuration, you configure an EtherChannel on each switch and then configure the vPC domain. You assign the EtherChannel on each switch as a peer link. For redundancy, we recommend that you should configure at least two dedicated ports into the EtherChannel; if one of the interfaces in the vPC peer link fails, the switch automatically falls back to use another interface in the peer link.

**Note**

---

We recommend that you configure the EtherChannels in trunk mode.

---

Many operational parameters and configuration parameters must be the same in each switch connected by a vPC peer link. Because each switch is completely independent on the management plane, you must ensure that the switches are compatible on the critical parameters. vPC peer switches have separate control planes. After configuring the vPC peer link, you should display the configuration on each vPC peer switch to ensure that the configurations are compatible.

**Note**

---

You must ensure that the two switches connected by the vPC peer link have certain identical operational and configuration parameters.

---

When you configure the vPC peer link, the vPC peer switches negotiate that one of the connected switches is the primary switch and the other connected switch is the secondary switch. By default, the Cisco NX-OS software uses the lowest MAC address to elect the primary switch. The software takes different actions on each switch—that is, the primary and secondary—only in certain failover conditions. If the primary switch fails, the secondary switch becomes the operational primary switch when the system recovers, and the previously primary switch is now the secondary switch.

You can also configure which of the vPC switches is the primary switch. If you want to configure the role priority again to make one vPC switch the primary switch, configure the role priority on both the primary and secondary vPC switches with the appropriate values, shut down the EtherChannel that is the vPC peer link on both switches by entering the **shutdown** command, and reenab the EtherChannel on both switches by entering the **no shutdown** command.

MAC addresses that are learned over vPC links are also synchronized between the peers.

Configuration information flows across the vPC peer links using the Cisco Fabric Services over Ethernet (CFSOE) protocol. All MAC addresses for those VLANs configured on both switches are synchronized between vPC peer switches. The software uses CFSOE for this synchronization.

If the vPC peer link fails, the software checks the status of the remote vPC peer switch using the peer-keepalive link, which is a link between vPC peer switches, to ensure that both switches are up. If the vPC peer switch is up, the secondary vPC switch disables all vPC ports on its switch. The data then forwards down the remaining active links of the EtherChannel.

The software learns of a vPC peer switch failure when the keepalive messages are not returned over the peer-keepalive link.

Use a separate link (vPC peer-keepalive link) to send configurable keepalive messages between the vPC peer switches. The keepalive messages on the vPC peer-keepalive link determines whether a failure is on the vPC peer link only or on the vPC peer switch. The keepalive messages are used only when all the links in the peer link fail.

## vPC Number

Once you have created the vPC domain ID and the vPC peer link, you can create EtherChannels to attach the downstream switch to each vPC peer switch. That is, you create one single EtherChannel on the downstream switch with half of the ports to the primary vPC peer switch and the other half of the ports to the secondary peer switch.

On each vPC peer switch, you assign the same vPC number to the EtherChannel that connects to the downstream switch. You will experience minimal traffic disruption when you are creating vPCs. To simplify the configuration, you can assign the vPC ID number for each EtherChannel to be the same as the EtherChannel itself (that is, vPC ID 10 for EtherChannel 10).

**Note**

The vPC number that you assign to the EtherChannel that connects to the downstream switch from the vPC peer switch must be identical on both vPC peer switches.

## vPC Interactions with Other Features

### Configuring vPC Peer Links and Links to the Core

Configure the command line interface by using a track object and a track list that is associated with the Layer 3 link to the core and on all vPC peer links on both vPC peer devices. You use this configuration to avoid dropping traffic if that particular module goes down because when all the tracked objects on the track list go down, the system does the following:

- Stops the vPC primary peer device sending peer-keepalive messages which forces the vPC secondary peer device to take over.
- Brings down all the downstream vPCs on that vPC peer device, which forces all the traffic to be rerouted in the access switch toward the other vPC peer device.

Once you configure this feature and if the module fails, the system automatically suspends all the vPC links on the primary vPC peer device and stops the peer-keepalive messages. This action forces the vPC secondary device to take over the primary role and all the vPC traffic to go to this new vPC primary device until the system stabilizes.

Create a track list that contains all the links to the core and all the vPC peer links as its object. Enable tracking for the specified vPC domain for this track list. Apply this same configuration to the other vPC peer device.

#### Before You Begin

To configure a track list to switch over vPC to the remote peer when all related interfaces fail:

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot/port*
3. switch(config-if)# **track track-id interface** *type slot/port line-protocol*
4. switch(config-track)# **track track-id interface** *type slot/port line-protocol*
5. switch(config)# **track track-id interface port-channel** *port line-protocol*
6. switch(config)# **track track-id list boolean** [OR | AND]
7. switch(config-track)# **object** *number*
8. switch(config-track)# **end**
9. switch(config)# **vpc domain** *domain-id*
10. switch(config-vpc-domain)# **track** *number*
11. (Optional) switch(config)# **show vpc brief**
12. (Optional) switch(config)# **copy running-config startup-config**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	switch(config)# <b>interface</b> <i>type slot/port</i>	Enters interface configuration mode.
Step 3	switch(config-if)# <b>track track-id interface</b> <i>type slot/port line-protocol</i>	Configures the track objects on an interface (Layer 3 to core).
Step 4	switch(config-track)# <b>track track-id interface</b> <i>type slot/port line-protocol</i>	Tracks the objects on an interface (Layer 3 to core).
Step 5	switch(config)# <b>track track-id interface port-channel</b> <i>port line-protocol</i>	Configures the track objects on a port channel (vPC peer link).
Step 6	switch(config)# <b>track track-id list boolean</b> [OR   AND]	Creates a track list that contains all the interfaces in the track list using the Boolean OR to trigger when all the objects fail, or trigger a switchover when any core interface or peer-link goes down using Boolean AND.
Step 7	switch(config-track)# <b>object</b> <i>number</i>	Specifies the object number.
Step 8	switch(config-track)# <b>end</b>	Exits track configuration mode.
Step 9	switch(config)# <b>vpc domain</b> <i>domain-id</i>	Enters vPC domain configuration.
Step 10	switch(config-vpc-domain)# <b>track</b> <i>number</i>	Adds the track object to the vPC domain.
Step 11	switch(config)# <b>show vpc brief</b>	(Optional) Displays the track object.

	Command or Action	Purpose
Step 12	switch(config)# <b>copy running-config startup-config</b>	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure a track list to trigger when all the objects fail using Boolean OR:

```
switch# configure terminal
switch(config)# interface ethernet 8/35
switch(config-if)# track 35 interface ethernet 8/35 line-protocol
switch(config-track)# track 23 interface ethernet 8/33 line-protocol
switch(config)# track 55 interface port-channel 100 line-protocol
switch(config)# track 44 list boolean OR
switch(config-track)# object 23
switch(config-track)# object 35
switch(config-track)# object 55
switch(config-track)# end
switch(config)# vpc domain 1
switch(config-vpc-domain)# track 44
switch(config)# copy running-config startup-config
```

## vPC and LACP

The Link Aggregation Control Protocol (LACP) uses the system MAC address of the vPC domain to form the LACP Aggregation Group (LAG) ID for the vPC.

You can use LACP on all the vPC EtherChannels, including those channels from the downstream switch. We recommend that you configure LACP with active mode on the interfaces on each EtherChannel on the vPC peer switches. This configuration allows you to more easily detect compatibility between switches, unidirectional links, and multihop connections, and provides dynamic reaction to run-time changes and link failures.

The vPC peer link supports 16 EtherChannel interfaces.



### Note

When you manually configure the system priority, you must ensure that you assign the same priority value on both vPC peer switches. If the vPC peer switches have different system priority values, vPC does not come up.



### Note

During a vPC role change (where secondary becomes primary), if the ports are in LACP-individual state on FEX, the ports will flap and this will cause brief connectivity loss before the primary is up. To avoid this issue, ensure that port-channel member ports are not configured as LACP-individual state or LACP is not used (On mode).

## vPC Peer Links and STP

When you first bring up the vPC functionality, STP reconverges. STP treats the vPC peer link as a special link and always includes the vPC peer link in the STP active topology.

We recommend that you set all the vPC peer link interfaces to the STP network port type so that Bridge Assurance is automatically enabled on all vPC peer links. We also recommend that you do not enable any of the STP enhancement features on VPC peer links.

You must configure a list of parameters to be identical on the vPC peer switches on both sides of the vPC peer link.

STP is distributed; that is, the protocol continues running on both vPC peer switches. However, the configuration on the vPC peer switch elected as the primary switch controls the STP process for the vPC interfaces on the secondary vPC peer switch.

The primary vPC switch synchronizes the STP state on the vPC secondary peer switch using Cisco Fabric Services over Ethernet (CFSoE).

The vPC manager performs a proposal/handshake agreement between the vPC peer switches that sets the primary and secondary switches and coordinates the two switches for STP. The primary vPC peer switch then controls the STP protocol for vPC interfaces on both the primary and secondary switches.

The Bridge Protocol Data Units (BPDUs) use the MAC address set for the vPC for the STP bridge ID in the designated bridge ID field. The vPC primary switch sends these BPDUs on the vPC interfaces.


**Note**

Display the configuration on both sides of the vPC peer link to ensure that the settings are identical. Use the **show spanning-tree** command to display information about the vPC.

## vPC and ARP

Table synchronization across vPC peers is managed in Cisco NX-OS using the reliable transport mechanism of the Cisco Fabric Services over Ethernet (CFSoE) protocol. To support faster convergence of address tables between the vPC peers, the **ip arp synchronize** command must be enabled. This convergence is designed to overcome the delay involved in ARP table restoration when the peer-link port channel flaps or when a vPC peer comes back online.

To improve performance, we recommend that you turn on the ARP sync feature. By default, it is not enabled.

To check whether or not ARP sync is enabled, enter the following command:

```
switch# show running
```

To enable ARP sync, enter the following command:

```
switch(config-vpc-domain) # ip arp synchronize
```

## CFSoE

The Cisco Fabric Services over Ethernet (CFSoE) is a reliable state transport mechanism that you can use to synchronize the actions of the vPC peer devices. CFSoE carries messages and packets for many features linked with vPC, such as STP and IGMP. Information is carried in CFS/CFSoE protocol data units (PDUs).

When you enable the vPC feature, the device automatically enables CFSoE, and you do not have to configure anything. CFSoE distributions for vPCs do not need the capabilities to distribute over IP or the CFS regions. You do not need to configure anything for the CFSoE feature to work correctly on vPCs.

You can use the **show mac address-table** command to display the MAC addresses that CFSoE synchronizes for the vPC peer link.

**Note**

Do not enter the **no cfs eth distribute** or the **no cfs distribute** command. CFSOE must be enabled for vPC functionality. If you do enter either of these commands when vPC is enabled, the system displays an error message.

When you enter the **show cfs application** command, the output displays "Physical-eth," which shows the applications that are using CFSOE.

## vPC Peer Switch

The vPC peer switch feature addresses performance concerns around STP convergence. This feature allows a pair of Cisco Nexus devices to appear as a single STP root in the Layer 2 topology. This feature eliminates the need to pin the STP root to the vPC primary switch and improves vPC convergence if the vPC primary switch fails.

To avoid loops, the vPC peer link is excluded from the STP computation. In vPC peer switch mode, STP BPDUs are sent from both vPC peer devices to avoid issues related to STP BPDU timeout on the downstream switches, which can cause traffic disruption.

This feature can be used with the pure peer switch topology in which the devices all belong to the vPC.

**Note**

Peer-switch feature is supported on networks that use vPC and STP-based redundancy is not supported. If the vPC peer-link fail in a hybrid peer-switch configuration, you can lose traffic. In this scenario, the vPC peers use the same STP root ID as well same bridge ID. The access switch traffic is split in two with half going to the first vPC peer and the other half to the second vPC peer. With the peer link failed, there is no impact on north/south traffic but east-west traffic will be lost (black-holed).

For information on STP enhancement features and Rapid PVST+, see the *Layer 2 Switching Configuration Guide* for your device.

## Guidelines and Limitations for vPCs

vPC has the following configuration guidelines and limitations:

- You must enable the vPC feature before you can configure vPC peer-link and vPC interfaces.
- You must configure the peer-keepalive link before the system can form the vPC peer link.
- The vPC peer-link needs to be formed using a minimum of two 10-Gigabit Ethernet interfaces.
- You can connect a pair of Cisco Nexus 5000 Series switches or a pair of Cisco Nexus 5500 Series switches in a vPC directly to another switch or to a server. vPC peer switches must be of the same type, for example, you can connect a pair of Nexus 5000 series switches or a pair of Nexus 5500 Series switches but you cannot connect a Nexus 5000 Series switch to a Nexus 5500 Series switch in a vPC topology.
- Only port channels can be in vPCs. A vPC can be configured on a normal port channel (switch-to-switch vPC topology), on a port channel fabric interface (fabric extender vPC topology), and on a port channel host interface (host interface vPC topology).

- A Fabric Extender can be a member of a Host Interface vPC topology or a Fabric Extender vPC topology but not both simultaneously.
- You must configure both vPC peer switches; the configuration is not automatically synchronized between the vPC peer devices.
- Check that the necessary configuration parameters are compatible on both sides of the vPC peer link.
- You may experience minimal traffic disruption while configuring vPCs.
- You should configure all the port channels in the vPC using LACP with the interfaces in active mode.
- When the **peer-switch** command is configured and vPC keepalive messages exchanged through an SVI instead of a management interface, additional Spanning Tree Protocol (STP) configuration is required. STP needs to be disabled on the dedicated link that carries the keepalive traffic between the vPC peers. You can disable STP on the dedicated link by configuring STP BPDUfilter on the both ends of the dedicated link. We recommend that the VLAN of the vPC keepalive SVI be allowed on only the interconnecting dedicated link and disallowed on all other links, including the peer link.
- You cannot have a link for non-vPC traffic in parallel with a vPC topology. This can cause errors with the traffic forwarding logic resulting in duplicate or missed packets.
- You cannot configure non-vPC interfaces across host ports on two different FEXs.

## Configuring vPCs

### Enabling vPCs

You must enable the vPC feature before you can configure and use vPCs.

#### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **feature vpc**
3. (Optional) switch# **show feature**
4. (Optional) switch# **copy running-config startup-config**

#### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>feature vpc</b>	Enables vPCs on the switch.
<b>Step 3</b>	switch# <b>show feature</b>	(Optional) Displays which features are enabled on the switch.

	Command or Action	Purpose
Step 4	switch# <b>copy running-config startup-config</b>	(Optional) Copies the running configuration to the startup configuration.

This example shows how to enable the vPC feature:

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

## Disabling vPCs

You can disable the vPC feature.



### Note

When you disable the vPC feature, the Cisco Nexus device clears all the vPC configurations.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **no feature vpc**
3. (Optional) switch# **show feature**
4. (Optional) switch# **copy running-config startup-config**

### DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	switch(config)# <b>no feature vpc</b>	Disables vPCs on the switch.
Step 3	switch# <b>show feature</b>	(Optional) Displays which features are enabled on the switch.
Step 4	switch# <b>copy running-config startup-config</b>	(Optional) Copies the running configuration to the startup configuration.

This example shows how to disable the vPC feature:

```
switch# configure terminal
switch(config)# no feature vpc
```

## Creating a vPC Domain

You must create identical vPC domain IDs on both the vPC peer devices. This domain ID is used to automatically form the vPC system MAC address.

### Before You Begin

Ensure that you have enabled the vPC feature.

You must configure both switches on either side of the vPC peer link.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vpc domain** *domain-id*
3. (Optional) switch# **show vpc brief**
4. (Optional) switch# **copy running-config startup-config**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>vpc domain</b> <i>domain-id</i>	Creates a vPC domain on the switch, and enters the vpc-domain configuration mode. There is no default <i>domain-id</i> ; the range is from 1 to 1000.  <b>Note</b> You can also use the <b>vpc domain</b> command to enter the vpc-domain configuration mode for an existing vPC domain.
<b>Step 3</b>	switch# <b>show vpc brief</b>	(Optional) Displays brief information about each vPC domain.
<b>Step 4</b>	switch# <b>copy running-config startup-config</b>	(Optional) Copies the running configuration to the startup configuration.

This example shows how to create a vPC domain:

```
switch# configure terminal
switch(config)# vpc domain 5
```

## Configuring a vPC Keepalive Link and Messages

You can configure the destination IP for the peer-keepalive link that carries the keepalive messages. Optionally, you can configure other parameters for the keepalive messages.

The Cisco NX-OS software uses the peer-keepalive link between the vPC peers to transmit periodic, configurable keepalive messages. You must have Layer 3 connectivity between the peer devices to transmit

these messages. The system cannot bring up the vPC peer link unless the peer-keepalive link is already up and running.

Ensure that both the source and destination IP addresses used for the peer-keepalive message are unique in your network and these IP addresses are reachable from the Virtual Routing and Forwarding (VRF) instance associated with the vPC peer-keepalive link.

**Note**

We recommend that you configure a separate VRF instance and put a Layer 3 port from each vPC peer switch into that VRF instance for the vPC peer-keepalive link. Do not use the peer link itself to send vPC peer-keepalive messages. For information on creating and configuring VRFs, see the Unicast Routing Configuration Guide for your device.

**Before You Begin**

Ensure that you have enabled the vPC feature.

You must configure the vPC peer-keepalive link before the system can form the vPC peer link.

You must configure both switches on either side of the vPC peer link.

**SUMMARY STEPS**

1. switch# **configure terminal**
2. switch(config)# **vpc domain** *domain-id*
3. switch(config-vpc-domain)# **peer-keepalive destination** *ipaddress* [**hold-timeout** *secs* | **interval** *msecs* {**timeout** *secs*} | **precedence** {*prec-value* | **network** | **internet** | **critical** | **flash-override** | **flash** | **immediate** | **priority** | **routine**} | **tos** {*tos-value* | **max-reliability** | **max-throughput** | **min-delay** | **min-monetary-cost** | **normal**} | **tos-byte** *tos-byte-value*} | **source** *ipaddress* | **vrf** {*name* | **management** | **vpc-keepalive**}]
4. (Optional) switch(config-vpc-domain)# **vpc peer-keepalive destination** *ipaddress* **source** *ipaddress*
5. (Optional) switch# **show vpc peer-keepalive**
6. (Optional) switch# **copy running-config startup-config**

**DETAILED STEPS**

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>vpc domain</b> <i>domain-id</i>	Creates a vPC domain on the switch if it does not already exist, and enters the vpc-domain configuration mode.
<b>Step 3</b>	switch(config-vpc-domain)# <b>peer-keepalive destination</b> <i>ipaddress</i> [ <b>hold-timeout</b> <i>secs</i>   <b>interval</b> <i>msecs</i> { <b>timeout</b> <i>secs</i> }   <b>precedence</b> { <i>prec-value</i>   <b>network</b>   <b>internet</b>   <b>critical</b>   <b>flash-override</b>   <b>flash</b>   <b>immediate</b>   <b>priority</b>   <b>routine</b> }   <b>tos</b> { <i>tos-value</i>   <b>max-reliability</b>   <b>max-throughput</b>   <b>min-delay</b>   <b>min-monetary-cost</b>   <b>normal</b> }   <b>tos-byte</b> <i>tos-byte-value</i> }   <b>source</b> <i>ipaddress</i>   <b>vrf</b> { <i>name</i>   <b>management</b>   <b>vpc-keepalive</b> }]	Configures the IPv4 address for the remote end of the vPC peer-keepalive link.  <b>Note</b> The system does not form the vPC peer link until you configure a vPC peer-keepalive link. The management ports and VRF are the defaults.

	Command or Action	Purpose
<b>Step 4</b>	switch(config-vpc-domain)# <b>vpc peer-keepalive destination ipaddress source ipaddress</b>	(Optional) Configures a separate VRF instance and puts a Layer 3 port from each vPC peer device into that VRF for the vPC peer-keepalive link.
<b>Step 5</b>	switch# <b>show vpc peer-keepalive</b>	(Optional) Displays information about the configuration for the keepalive messages.
<b>Step 6</b>	switch# <b>copy running-config startup-config</b>	(Optional) Copies the running configuration to the startup configuration.

This example shows how to configure the destination IP address for the vPC-peer-keepalive link:

```
switch# configure terminal
switch(config)# vpc domain 5
switch(config-vpc-domain)# peer-keepalive destination 10.10.10.42
```

This example shows how to set up the peer keepalive link connection between the primary and secondary vPC device:

```
switch(config)# vpc domain 100
switch(config-vpc-domain)# peer-keepalive destination 192.168.2.2 source 192.168.2.1
Note:-----: Management VRF will be used as the default VRF ::-----
switch(config-vpc-domain)#
```

This example shows how to create a separate VRF named vpc\_keepalive for the vPC keepalive link and how to verify the new VRF:

```
vrf context vpc_keepalive
interface Ethernet1/31
  switchport access vlan 123
interface Vlan123
  vrf member vpc_keepalive
  ip address 123.1.1.2/30
  no shutdown
vpc domain 1
  peer-keepalive destination 123.1.1.1 source 123.1.1.2 vrf
  vpc_keepalive

L3-NEXUS-2# show vpc peer-keepalive

vPC keep-alive status           : peer is alive
--Peer is alive for             : (154477) seconds, (908) msec
--Send status                   : Success
--Last send at                  : 2011.01.14 19:02:50 100 ms
--Sent on interface             : Vlan123
--Receive status                : Success
--Last receive at               : 2011.01.14 19:02:50 103 ms
--Received on interface         : Vlan123
--Last update from peer        : (0) seconds, (524) msec

vPC Keep-alive parameters
--Destination                   : 123.1.1.1
--Keepalive interval            : 1000 msec
--Keepalive timeout             : 5 seconds
```

```
--Keepalive hold timeout      : 3 seconds
--Keepalive vrf               : vpc_keepalive
--Keepalive udp port         : 3200
--Keepalive tos               : 192
```

The services provided by the switch, such as ping, ssh, telnet, radius, are VRF aware. The VRF name need to be configured or specified in order for the correct routing table to be used.

```
L3-NEXUS-2# ping 123.1.1.1 vrf vpc_keepalive
PING 123.1.1.1 (123.1.1.1): 56 data bytes
64 bytes from 123.1.1.1: icmp_seq=0 ttl=254 time=3.234 ms
64 bytes from 123.1.1.1: icmp_seq=1 ttl=254 time=4.931 ms
64 bytes from 123.1.1.1: icmp_seq=2 ttl=254 time=4.965 ms
64 bytes from 123.1.1.1: icmp_seq=3 ttl=254 time=4.971 ms
64 bytes from 123.1.1.1: icmp_seq=4 ttl=254 time=4.915 ms
```

```
--- 123.1.1.1 ping statistics ---
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min/avg/max = 3.234/4.603/4.971 ms
```

## Creating a vPC Peer Link

You can create a vPC peer link by designating the EtherChannel that you want on each switch as the peer link for the specified vPC domain. We recommend that you configure the EtherChannels that you are designating as the vPC peer link in trunk mode and that you use two ports on separate modules on each vPC peer switch for redundancy.

### Before You Begin

Ensure that you have enabled the vPC feature.

You must configure both switches on either side of the vPC peer link

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface port-channel** *channel-number*
3. switch(config-if)# **vpc peer-link**
4. (Optional) switch# **show vpc brief**
5. (Optional) switch# **copy running-config startup-config**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface port-channel</b> <i>channel-number</i>	Selects the EtherChannel that you want to use as the vPC peer link for this switch, and enters the interface configuration mode.
<b>Step 3</b>	switch(config-if)# <b>vpc peer-link</b>	Configures the selected EtherChannel as the vPC peer link, and enters the vpc-domain configuration mode.

	Command or Action	Purpose
Step 4	switch# <b>show vpc brief</b>	(Optional) Displays information about each vPC, including information about the vPC peer link.
Step 5	switch# <b>copy running-config startup-config</b>	(Optional) Copies the running configuration to the startup configuration.

This example shows how to configure a vPC peer link:

```
switch# configure terminal
switch(config)# interface port-channel 20
switch(config-if)# vpc peer-link
```

## Checking the Configuration Compatibility

After you have configured the vPC peer link on both vPC peer switches, check that the configurations are consistent on all vPC interfaces.



### Note

Beginning with Cisco NX-OS Release 5.0(2)N1(1), the The following QoS parameters support Type 2 consistency checks:

- Network QoS—MTU and Pause
- Input Queuing —Bandwidth and Absolute Priority
- Output Queuing—Bandwidth and Absolute Priority

In the case of a Type 2 mismatch, the vPC is not suspended. Type 1 mismatches suspend the vPC.

Command or Action	Purpose
switch# <b>show vpc consistency-parameters {global   interface port-channel channel-number}</b>	Displays the status of those parameters that must be consistent across all vPC interfaces.

This example shows how to check that the required configurations are compatible across all the vPC interfaces:

```
switch# show vpc consistency-parameters global
Legend:
Type 1 : vPC will be suspended in case of mismatch
Name                Type  Local Value                Peer Value
-----
QoS                  2      ([], [], [], [], [],  ([], [], [], [], [],
                    [])
Network QoS (MTU)   2      (1538, 0, 0, 0, 0, 0)  (1538, 0, 0, 0, 0, 0)
Network QoS (Pause) 2      (F, F, F, F, F, F)    (1538, 0, 0, 0, 0, 0)
Input Queuing (Bandwidth) 2      (100, 0, 0, 0, 0, 0)  (100, 0, 0, 0, 0, 0)
Input Queuing (Absolute Priority) 2      (F, F, F, F, F, F)    (100, 0, 0, 0, 0, 0)
Output Queuing (Bandwidth) 2      (100, 0, 0, 0, 0, 0)  (100, 0, 0, 0, 0, 0)
```

```

Output Queuing (Absolute Priority) 2 (F, F, F, F, F, F) (100, 0, 0, 0, 0, 0)
STP Mode 1 Rapid-PVST Rapid-PVST
STP Disabled 1 None None
STP MST Region Name 1 "" ""
STP MST Region Revision 1 0 0
STP MST Region Instance to VLAN Mapping 1
STP Loopguard 1 Disabled Disabled
STP Bridge Assurance 1 Enabled Enabled
STP Port Type, Edge 1 Normal, Disabled, Normal, Disabled,
BPDUFILTER, Edge BPDUGuard 1 Disabled Disabled
STP MST Simulate PVST 1 Enabled Enabled
Allowed VLANs - 1,624 1
Local suspended VLANs switch# - 624 -
    
```

This example shows how to check that the required configurations are compatible for an EtherChannel interface:

```
switch# show vpc consistency-parameters interface port-channel 20
```

Legend:

Type 1 : vPC will be suspended in case of mismatch

Name	Type	Local Value	Peer Value
Fex id	1	20	20
STP Port Type	1	Default	Default
STP Port Guard	1	None	None
STP MST Simulate PVST	1	Default	Default
mode	1	on	on
Speed	1	10 Gb/s	10 Gb/s
Duplex	1	full	full
Port Mode	1	fex-fabric	fex-fabric
Shut Lan	1	No	No
Allowed VLANs	-	1,3-3967,4048-4093	1-3967,4048-4093

## Enabling vPC Auto-Recovery

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vpc domain** *domain-id*
3. switch(config-vpc-domain)# **auto-recovery reload-delay** *delay*

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>vpc domain</b> <i>domain-id</i>	Enters vpc-domain configuration mode for an existing vPC domain.
<b>Step 3</b>	switch(config-vpc-domain)# <b>auto-recovery reload-delay</b> <i>delay</i>	Enables the auto-recovery feature and sets the reload delay period. The default is disabled.

This example shows how to enable the auto-recovery feature in vPC domain 10 and set the delay period for 240 seconds:

```
switch(config)# vpc domain 10
switch(config-vpc-domain)# auto-recovery reload-delay 240
Warning:
  Enables restoring of vPCs in a peer-detached state after reload, will wait for 240 seconds
  (by default) to determine if peer is un-reachable
```

This example shows how to view the status of the auto-recovery feature in vPC domain 10:

```
switch(config-vpc-domain)# show running-config vpc
!Command: show running-config vpc
!Time: Tue Dec 7 02:38:44 2010

version 5.0(2)N2(1)

feature vpc
vpc domain 10
  peer-keepalive destination 10.193.51.170
  auto-recovery
```

## Configuring the Restore Time Delay

Beginning with Cisco NX-OS Release 5.0(3)N1(1), you can configure a restore timer that delays the vPC from coming back up until after the peer adjacency forms and the VLAN interfaces are back up. This feature avoids packet drops if the routing tables fail to converge before the vPC is once again passing traffic.

### Before You Begin

Ensure that you have enabled the vPC feature.

You must configure both switches on either side of the vPC peer link with the following procedures.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vpc domain** *domain-id*
3. switch(config-vpc-domain)# **delay restore** *time*
4. (Optional) switch# **copy running-config startup-config**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>vpc domain</b> <i>domain-id</i>	Creates a vPC domain on the switch if it does not already exist, and enters vpc-domain configuration mode.
<b>Step 3</b>	switch(config-vpc-domain)# <b>delay restore</b> <i>time</i>	Configures the time delay before the vPC is restored.  The restore time is the number of seconds to delay bringing up the restored vPC peer device. The range is from 1 to 3600. The default is 30 seconds.

	Command or Action	Purpose
Step 4	switch# <b>copy running-config startup-config</b>	(Optional) Copies the running configuration to the startup configuration.

This example shows how to configure the delay reload time for a vPC link:

```
switch(config)# vpc domain 1
switch(config-vpc-domain)# delay restore 10
switch(config-vpc-domain)#
```

## Excluding VLAN Interfaces from Shutting Down a vPC Peer Link Fails

When a vPC peer-link is lost, the vPC secondary switch suspends its vPC member ports and its switch virtual interface (SVI) interfaces. All Layer 3 forwarding is disabled for all VLANs on the vPC secondary switch. You can exclude specific SVI interfaces so that they are not suspended.

### Before You Begin

Ensure that the VLAN interfaces have been configured.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vpc domain** *domain-id*
3. switch(config-vpc-domain)# **dual-active exclude interface-vlan** *range*

### DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	switch(config)# <b>vpc domain</b> <i>domain-id</i>	Creates a vPC domain on the switch if it does not already exist, and enters vpc-domain configuration mode.
Step 3	switch(config-vpc-domain)# <b>dual-active exclude interface-vlan</b> <i>range</i>	Specifies the VLAN interfaces that should remain up when a vPC peer-link is lost.  range—Range of VLAN interfaces that you want to exclude from shutting down. The range is from 1 to 4094.

This example shows how to keep the interfaces on VLAN 10 up on the vPC peer switch if a peer link fails:

```
switch# configure terminal
switch(config)# vpc domain 5
switch(config-vpc-domain)# dual-active exclude interface-vlan 10
switch(config-vpc-domain)#
```

## Configuring the VRF Name

The switch services, such as ping, ssh, telnet, radius, are VRF aware. You must configure the VRF name in order for the correct routing table to be used.

You can specify the VRF name.

### SUMMARY STEPS

1. switch# **ping** *ipaddress* **vrf** *vrf-name*

### DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# <b>ping</b> <i>ipaddress</i> <b>vrf</b> <i>vrf-name</i>	Specifies the virtual routing and forwarding (VRF) name to use. The VRF name is case sensitive and can be a maximum of 32 characters..

This example shows how to specify the VRF named vpc\_keepalive:

```
switch# ping 123.1.1.1 vrf vpc_keepalive
PING 123.1.1.1 (123.1.1.1): 56 data bytes
64 bytes from 123.1.1.1: icmp_seq=0 ttl=254 time=3.234 ms
64 bytes from 123.1.1.1: icmp_seq=1 ttl=254 time=4.931 ms
64 bytes from 123.1.1.1: icmp_seq=2 ttl=254 time=4.965 ms
64 bytes from 123.1.1.1: icmp_seq=3 ttl=254 time=4.971 ms
64 bytes from 123.1.1.1: icmp_seq=4 ttl=254 time=4.915 ms

--- 123.1.1.1 ping statistics ---
5 packets transmitted, 5 packets received, 0.00% packet loss
round-trip min/avg/max = 3.234/4.603/4.971 ms
```

## Binding a VRF Instance to a vPC

You can bind a VRF instance to a vPC. One reserved VLAN is required for each VRF. Without this command, the receivers in a non-VPC VLAN and the receivers connected to a Layer 3 interface might not receive multicast traffic. The non-vPC VLANs are the VLANs that are not trunked over a peerlink.

### Before You Begin

Use the **show interfaces brief** command to view the interfaces that are in use on a switch. To bind the VRF instance to the vPC, you must use a VLAN that is not already in use.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vpc bind-vrf** *vrf-name* **vlan** *vlan-id*

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>vpc bind-vrf vrf-name vlan vlan-id</b>	Binds a VRF instance to a vPC and specifies the VLAN to bind to the vPC. The VLAN ID range is from 1 to 3967 and from 4049 to 4093.

This example shows how to bind a vPC to the default VRF instance using VLAN 2:

```
switch(config)# vpc bind-vrf default vlan vlan2
```

## Enabling Layer 3 Forwarding to the Gateway MAC Address of the vPC

The vPC peer-gateway feature allows a vPC switch to act as the active gateway for packets that are addressed to the router MAC address of the vPC peer. You can enable local forwarding without the need to cross the vPC peer-link. In this scenario, the feature optimizes use of the peer-link and avoids potential traffic loss.

You can enable Layer 3 forwarding for packets destined to the gateway MAC address of the virtual Port Channel (vPC).

**Note**

You must configure this feature on both vPC peer switches.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vpc domain domain-id**
3. switch(config-vpc-domain)# **peer-gateway range**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters configuration mode.
<b>Step 2</b>	switch(config)# <b>vpc domain domain-id</b>	Creates a vPC domain on the switch if it does not already exist, and enters the vpc-domain configuration mode.
<b>Step 3</b>	switch(config-vpc-domain)# <b>peer-gateway range</b>	Enables Layer 3 forwarding for packets destined to the gateway MAC address of the virtual Port Channel (vPC).

This example shows how to enable the vPC peer gateway:

```
switch(config)# vpc domain 20
switch(config-vpc-domain)# peer-gateway
switch(config-vpc-domain)#
```

## Suspending Orphan Ports on a Secondary Switch in a vPC Topology

You can suspend a non-virtual port channel (vPC) port when a vPC secondary peer link goes down. A non-vPC port, also known as an orphaned port, is a port that is not part of a vPC.



### Note

When a port is configured as an orphan port, the port will flap. This occurs because the system reevaluates whether the port can be brought up, given the constraints of the orphan port. For example, MCT needs to be up and election needs to be complete.

### Before You Begin

Enable the vPC feature.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface ethernet slot/port**
3. switch(config-if)# **vpc orphan-port suspend**
4. switch(config-if)# **exit**
5. (Optional) switch# **show vpc orphan-port**
6. (Optional) switch(config-if)# **copy running-config startup-config**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface ethernet slot/port</b>	Specifies the port that you want to configure and enters interface configuration mode.
<b>Step 3</b>	switch(config-if)# <b>vpc orphan-port suspend</b>	Suspends the specified port if the secondary switch goes down. <b>Note</b> The <b>vpc-orphan-port suspend</b> command is supported only on physical ports, and not supported on port-channel member ports.
<b>Step 4</b>	switch(config-if)# <b>exit</b>	Exits interface configuration mode.
<b>Step 5</b>	switch# <b>show vpc orphan-port</b>	(Optional) Displays the orphan port configuration.

	Command or Action	Purpose
Step 6	switch(config-if)# <b>copy running-config startup-config</b>	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to suspend an orphan port:

```
switch# configure terminal
switch(config)# interface ethernet 1/0
switch(config-if)# vpc orphan-port suspend
```

This example shows how to display ports that are not part of the vPC but that share common VLANs with ports that are part of the vPC:

```
switch# configure terminal
switch(config)# show vpc orphan-ports
Note:
-----:::Going through port database. Please be patient.:::-----
VLAN Orphan Ports
-----
 1 Po600
 2 Po600
 3 Po600
 4 Po600
 5 Po600
 6 Po600
 7 Po600
 8 Po600
 9 Po600
10 Po600
11 Po600
12 Po600
13 Po600
14 Po600
...
```

## Creating an EtherChannel Host Interface

To connect to a downstream server from a Cisco Nexus Fabric Extender you can create a EtherChannel host interface. An EtherChannel host interface can have only one host interface as a member depending on the fabric extender model. The Cisco Nexus 2148T allows only one interface member per fabric extender, newer fabric extenders allow up to 8 members of the same port-channel on a single fabric extender. You need to create an EtherChannel host interface to configure a vPC on it that uses the Fabric Extender topology.

### Before You Begin

Ensure that you have enabled the vPC feature.

Ensure that the connected Fabric Extender is online.

You must configure both switches on either side of the vPC peer link with the following procedure.

**SUMMARY STEPS**

1. switch# **configure terminal**
2. switch(config)# **interface ethernet** *chassis/slot/port*
3. switch(config-if)# **channel-group** *channel-number* **mode** {**active** | **passive** | **on**}
4. (Optional) switch(config-if)# **show port-channel summary**
5. (Optional) switch(config-if)# **copy running-config startup-config**

**DETAILED STEPS**

	<b>Command or Action</b>	<b>Purpose</b>
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface ethernet</b> <i>chassis/slot/port</i>	Specifies an interface to configure, and enters interface configuration mode.
<b>Step 3</b>	switch(config-if)# <b>channel-group</b> <i>channel-number</i> <b>mode</b> { <b>active</b>   <b>passive</b>   <b>on</b> }	Creates an EtherChannel host interface on the selected host interface.
<b>Step 4</b>	switch(config-if)# <b>show port-channel summary</b>	(Optional) Displays information about each EtherChannel host interface.
<b>Step 5</b>	switch(config-if)# <b>copy running-config startup-config</b>	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure an EtherChannel host interface:

```
switch# configure terminal
switch(config)# interface ethernet 101/1/20
switch(config-if)# channel-group 7 mode active
```

**Moving Other Port Channels into a vPC****Before You Begin**

Ensure that you have enabled the vPC feature.

You must configure both switches on either side of the vPC peer link with the following procedure.

**SUMMARY STEPS**

1. switch# **configure terminal**
2. switch(config)# **interface port-channel** *channel-number*
3. switch(config-if)# **vpc** *number*
4. (Optional) switch# **show vpc brief**
5. (Optional) switch# **copy running-config startup-config**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface port-channel</b> <i>channel-number</i>	Selects the port channel that you want to put into the vPC to connect to the downstream switch, and enters interface configuration mode.  <b>Note</b> A vPC can be configured on a normal port channel (physical vPC topology), on an port channel fabric interface (fabric extender vPC topology), and on a port channel host interface (host interface vPC topology)
<b>Step 3</b>	switch(config-if)# <b>vpc</b> <i>number</i>	Configures the selected port channel into the vPC to connect to the downstream switch. The range is from 1 to 4096.  The vPC <i>number</i> that you assign to the port channel that connects to the downstream switch from the vPC peer switch must be identical on both vPC peer switches.
<b>Step 4</b>	switch# <b>show vpc brief</b>	(Optional) Displays information about each vPC.
<b>Step 5</b>	switch# <b>copy running-config startup-config</b>	(Optional) Copies the running configuration to the startup configuration.

This example shows how to configure a port channel that will connect to the downstream device:

```
switch# configure terminal
switch(config)# interface port-channel 20
switch(config-if)# vpc 5
```

## Manually Configuring a vPC Domain MAC Address



**Note** Configuring the system address is an optional configuration step.

### Before You Begin

Ensure that you have enabled the vPC feature.

You must configure both switches on either side of the vPC peer link.

**SUMMARY STEPS**

1. switch# **configure terminal**
2. switch(config)# **vpc domain** *domain-id*
3. switch(config-vpc-domain)# **system-mac** *mac-address*
4. (Optional) switch# **show vpc role**
5. (Optional) switch# **copy running-config startup-config**

**DETAILED STEPS**

	<b>Command or Action</b>	<b>Purpose</b>
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>vpc domain</b> <i>domain-id</i>	Selects an existing vPC domain on the switch, or creates a new vPC domain, and enters the vpc-domain configuration mode. There is no default <i>domain-id</i> ; the range is from 1 to 1000.
<b>Step 3</b>	switch(config-vpc-domain)# <b>system-mac</b> <i>mac-address</i>	Enters the MAC address that you want for the specified vPC domain in the following format: aaaa.bbbb.cccc.
<b>Step 4</b>	switch# <b>show vpc role</b>	(Optional) Displays the vPC system MAC address.
<b>Step 5</b>	switch# <b>copy running-config startup-config</b>	(Optional) Copies the running configuration to the startup configuration.

This example shows how to configure a vPC domain MAC address:

```
switch# configure terminal
switch(config)# vpc domain 5
switch(config-if)# system-mac 23fb.4ab5.4c4e
```

**Manually Configuring the System Priority**

When you create a vPC domain, the system automatically creates a vPC system priority. However, you can also manually configure a system priority for the vPC domain.

**Before You Begin**

Ensure that you have enabled the vPC feature.

You must configure both switches on either side of the vPC peer link.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vpc domain** *domain-id*
3. switch(config-vpc-domain)# **system-priority** *priority*
4. (Optional) switch# **show vpc brief**
5. (Optional) switch# **copy running-config startup-config**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>vpc domain</b> <i>domain-id</i>	Selects an existing vPC domain on the switch, or creates a new vPC domain, and enters the vpc-domain configuration mode. There is no default <i>domain-id</i> ; the range is from 1 to 1000.
<b>Step 3</b>	switch(config-vpc-domain)# <b>system-priority</b> <i>priority</i>	Enters the system priority that you want for the specified vPC domain. The range of values is from 1 to 65535. The default value is 32667.
<b>Step 4</b>	switch# <b>show vpc brief</b>	(Optional) Displays information about each vPC, including information about the vPC peer link.
<b>Step 5</b>	switch# <b>copy running-config startup-config</b>	(Optional) Copies the running configuration to the startup configuration.

This example shows how to configure a vPC peer link:

```
switch# configure terminal
switch(config)# vpc domain 5
switch(config-if)# system-priority 4000
```

## Manually Configuring a vPC Peer Switch Role

By default, the Cisco NX-OS software elects a primary and secondary vPC peer switch after you configure the vPC domain and both sides of the vPC peer link. However, you may want to elect a specific vPC peer switch as the primary switch for the vPC. Then, you would manually configure the role value for the vPC peer switch that you want as the primary switch to be lower than the other vPC peer switch.

vPC does not support role preemption. If the primary vPC peer switch fails, the secondary vPC peer switch takes over to become operationally the vPC primary switch. However, the original operational roles are not restored when the formerly primary vPC comes up again.

### Before You Begin

Ensure that you have enabled the vPC feature.

You must configure both switches on either side of the vPC peer link.

**SUMMARY STEPS**

1. switch# **configure terminal**
2. switch(config)# **vpc domain** *domain-id*
3. switch(config-vpc-domain)# **role priority** *priority*
4. (Optional) switch# **show vpc brief**
5. (Optional) switch# **copy running-config startup-config**

**DETAILED STEPS**

	<b>Command or Action</b>	<b>Purpose</b>
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>vpc domain</b> <i>domain-id</i>	Selects an existing vPC domain on the switch, or creates a new vPC domain, and enters the vpc-domain configuration mode. There is no default <i>domain-id</i> ; the range is from 1 to 1000.
<b>Step 3</b>	switch(config-vpc-domain)# <b>role priority</b> <i>priority</i>	Enters the role priority that you want for the vPC system priority. The range of values is from 1 to 65535. The default value is 32667.
<b>Step 4</b>	switch# <b>show vpc brief</b>	(Optional) Displays information about each vPC, including information about the vPC peer link.
<b>Step 5</b>	switch# <b>copy running-config startup-config</b>	(Optional) Copies the running configuration to the startup configuration.

This example shows how to configure a vPC peer link:

```
switch# configure terminal
switch(config)# vpc domain 5
switch(config-if)# role priority 4000
```

## Configuring the vPC Peer Switch

### Configuring a Pure vPC Peer Switch Topology

You can configure a pure vPC peer switch topology using the **peer-switch** command and then you set the best possible (lowest) spanning tree bridge priority value.

**Note**

The values you apply for the spanning tree priority must be identical on both vPC peers.

### Before You Begin

Ensure that you have enabled the vPC feature.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vpc domain** *domain-id*
3. switch(config-vpc-domain)# **peer-switch**
4. switch(config-vpc-domain)# **spanning-tree vlan** *vlan-range* **priority value**
5. switch(config-vpn-domain)# **exit**
6. (Optional) switch(config)# **show spanning-tree summary**
7. (Optional) switch(config)# **copy running-config startup-config**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>vpc domain</b> <i>domain-id</i>	Enters the vPC domain number that you want to configure. The system enters the vpc-domain configuration mode.
<b>Step 3</b>	switch(config-vpc-domain)# <b>peer-switch</b>	Enables the vPC switch pair to appear as a single STP root in the Layer 2 topology.  Use the <b>no</b> form of the command to disable the peer switch vPC topology.
<b>Step 4</b>	switch(config-vpc-domain)# <b>spanning-tree vlan</b> <i>vlan-range</i> <b>priority value</b>	Configures the bridge priority of the VLAN. Valid values are multiples of 4096. The default value is 32768.  <b>Note</b> This value must be identical on both vPC peers.
<b>Step 5</b>	switch(config-vpn-domain)# <b>exit</b>	Exits the vpc-domain configuration mode.
<b>Step 6</b>	switch(config)# <b>show spanning-tree summary</b>	(Optional) Displays a summary of the spanning tree port states including the vPC peer switch.  Look for the following line in the command output: vPC peer-switch is enabled (operational)
<b>Step 7</b>	switch(config)# <b>copy running-config startup-config</b>	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure a pure vPC peer switch topology:

```
switch# configure terminal
```

```

Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# vpc domain 5
switch(config-vpc-domain)# peer-switch
2010 Apr 28 14:44:44 switch %STP-2-VPC_PEERSWITCH_CONFIG_ENABLED: vPC peer-switch
configuration is enabled. Please make sure to configure spanning tree "bridge" priority as
per recommended guidelines to make vPC peer-switch operational.
switch(config-vpc-domain)# exit
switch(config)# spanning-tree vlan 1 priority 8192
switch(config)# show spanning-tree summary
Switch is in rapid-pvst mode
Root bridge for: VLAN0001-VLAN0050, VLAN0100-VLAN0149, VLAN0200-VLAN0249
VLAN0300-VLAN0349, VLAN0400-VLAN0599, VLAN0900-VLAN0999
Port Type Default is disable
Edge Port [PortFast] BPDU Guard Default is disabled
Edge Port [PortFast] BPDU Filter Default is disabled
Bridge Assurance is enabled
Loopguard Default is disabled
Pathcost method used is short
vPC peer-switch is enabled (operational)
Name Blocking Listening Learning Forwarding STP Active
-----
VLAN0001 0 0 0 16 16
VLAN0002 0 0 0 16 16
switch(config)# copy running-config startup-config
switch(config)#

```

## Configuring a Hybrid vPC Peer Switch Topology

You can configure a hybrid vPC and non-vPC peer switch topology by using the spanning-tree pseudo-information command to change the designated bridge ID so that it meets the STP VLAN-based load-balancing criteria and then change the root bridge ID priority to a value that is better than the best bridge priority. You then enable the peer switch. For more information, see the command reference for your device.



### Note

If you previously configured global spanning tree parameters and you subsequently configure spanning tree pseudo information parameters, be aware that the pseudo information parameters take precedence over the global parameters.

### Before You Begin

Ensure that you have enabled the vPC feature.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree pseudo-information**
3. switch(config-pseudo)# **vlan *vlan-id* designated priority *priority***
4. switch(config-pseudo)# **vlan *vlan-id* root priority *priority***
5. switch(config-pseudo)# **exit**
6. switch(config)# **vpc domain *domain-id***
7. switch(config-vpc-domain)# **peer-switch**
8. switch(config-vpc-domain)# **exit**
9. (Optional) switch(config)# **show spanning-tree summary**
10. (Optional) switch(config)# **copy running-config startup-config**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>spanning-tree pseudo-information</b>	Configures the spanning tree pseudo information. <b>Note</b> This configuration takes precedence over any global spanning tree configurations.
<b>Step 3</b>	switch(config-pseudo)# <b>vlan vlan-id designated priority priority</b>	Configures the designated bridge priority of the VLAN. Valid values are multiples of 4096 from 0 to 61440.
<b>Step 4</b>	switch(config-pseudo)# <b>vlan vlan-id root priority priority</b>	Configures the root bridge priority of the VLAN. Valid values are multiples of 4096 from 0 to 61440. <b>Note</b> This value must be identical on both vPC peers to have an operational peer switch.
<b>Step 5</b>	switch(config-pseudo)# <b>exit</b>	Exits spanning tree pseudo information configuration mode.
<b>Step 6</b>	switch(config)# <b>vpc domain domain-id</b>	Enters the vPC domain number that you want to configure. The system enters the vpc-domain configuration mode.
<b>Step 7</b>	switch(config-vpc-domain)# <b>peer-switch</b>	Enables the vPC switch pair to appear as a single STP root in the Layer 2 topology. Use the <b>no</b> form of the command to disable the peer switch vPC topology.
<b>Step 8</b>	switch(config-vpc-domain)# <b>exit</b>	Exits the vpc-domain configuration mode.
<b>Step 9</b>	switch(config)# <b>show spanning-tree summary</b>	(Optional) Displays a summary of the spanning tree port states including the vPC peer switch. Look for the following line in the command output: vPC peer-switch is enabled (operational)
<b>Step 10</b>	switch(config)# <b>copy running-config startup-config</b>	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure a hybrid vPC peer switch topology:

```
switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# spanning-tree pseudo-information
switch(config-pseudo)# vlan 1 designated priority 8192
switch(config-pseudo)# vlan 1 root priority 4096
switch(config-pseudo)# exit
switch(config)# vpc domain 5
switch(config-vpc-domain)# peer-switch
switch(config-vpc-domain)# exit
switch(config)# copy running-config startup-config
```

## Verifying the vPC Configuration

Use the following commands to display vPC configuration information:

Command	Purpose
switch# <b>show feature</b>	Displays whether vPC is enabled or not.
switch# <b>show port-channel capacity</b>	Displays how many EtherChannels are configured and how many are still available on the switch.
switch# <b>show running-config vpc</b>	Displays running configuration information for vPCs.
switch# <b>show vpc brief</b>	Displays brief information on the vPCs.
switch# <b>show vpc consistency-parameters</b>	Displays the status of those parameters that must be consistent across all vPC interfaces.
switch# <b>show vpc peer-keepalive</b>	Displays information on the peer-keepalive messages.
switch# <b>show vpc role</b>	Displays the peer status, the role of the local switch, the vPC system MAC address and system priority, and the MAC address and priority for the local vPC switch.
switch# <b>show vpc statistics</b>	Displays statistics on the vPCs.  <b>Note</b> This command displays the vPC statistics only for the vPC peer device that you are working on.

For information about the switch output, see the Command Reference for your Cisco Nexus Series switch.

## Viewing the Graceful Type-1 Check Status

This example shows how to display the current status of the graceful Type-1 consistency check:

```
switch# show vpc brief
Legend:
          (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id           : 10
Peer status              : peer adjacency formed ok
vPC keep-alive status   : peer is alive
Configuration consistency status: success
Per-vlan consistency status : success
Type-2 consistency status : success
vPC role                 : secondary
Number of vPCs configured : 34
Peer Gateway             : Disabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Enabled

vPC Peer-link status
```

```

-----
id   Port   Status Active vlans
--   -
1    Po1    up     1
-----

```

## Viewing a Global Type-1 Inconsistency

When a global Type-1 inconsistency occurs, the vPCs on the secondary switch are brought down. The following example shows this type of inconsistency when there is a spanning-tree mode mismatch.

The example shows how to display the status of the suspended vPC VLANs on the secondary switch:

```

switch(config)# show vpc
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id           : 10
Peer status             : peer adjacency formed ok
vPC keep-alive status   : peer is alive
Configuration consistency status: failed
Per-vlan consistency status : success
Configuration consistency reason: vPC type-1 configuration incompatible - STP
                                Mode inconsistent

Type-2 consistency status : success
vPC role                 : secondary
Number of vPCs configured : 2
Peer Gateway             : Disabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Enabled

vPC Peer-link status
-----
id   Port   Status Active vlans
--   -
1    Po1    up     1-10
-----

vPC status
-----
id   Port   Status Consistency Reason Active vlans
--   -
20   Po20   down* failed Global compat check failed -
30   Po30   down* failed Global compat check failed -
-----

```

The example shows how to display the inconsistent status ( the VLANs on the primary vPC are not suspended) on the primary switch:

```

switch(config)# show vpc
Legend:
                (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id           : 10
Peer status             : peer adjacency formed ok
vPC keep-alive status   : peer is alive
Configuration consistency status: failed
Per-vlan consistency status : success
Configuration consistency reason: vPC type-1 configuration incompatible - STP Mo
de inconsistent
Type-2 consistency status : success
vPC role                 : primary
Number of vPCs configured : 2
Peer Gateway             : Disabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Enabled

vPC Peer-link status
-----
id   Port   Status Active vlans
--   -
1    Po1    up     1-10
-----

```

```
vPC status
-----
id      Port      Status Consistency Reason                Active vlans
-----
20      Po20      up     failed    Global compat check failed 1-10
30      Po30      up     failed    Global compat check failed 1-10
```

## Viewing an Interface-Specific Type-1 Inconsistency

When an interface-specific Type-1 inconsistency occurs, the vPC port on the secondary switch is brought down while the primary switch vPC ports remain up. The following example shows this type of inconsistency when there is a switchport mode mismatch.

This example shows how to display the status of the suspended vPC VLAN on the secondary switch:

```
switch(config-if)# show vpc brief
Legend:
      (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id          : 10
Peer status             : peer adjacency formed ok
vPC keep-alive status  : peer is alive
Configuration consistency status: success
Per-vlan consistency status : success
Type-2 consistency status : success
vPC role                : secondary
Number of vPCs configured : 2
Peer Gateway            : Disabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Enabled

vPC Peer-link status
-----
id  Port  Status Active vlans
--  ---  -
1   Po1   up     1

vPC status
-----
id      Port      Status Consistency Reason                Active vlans
-----
20      Po20      up     success    success                1
30      Po30      down*  failed    Compatibility check failed -
                                     for port mode
```

This example shows how to display the inconsistent status (the VLANs on the primary vPC are not suspended) on the primary switch:

```
switch(config-if)# show vpc brief
Legend:
      (*) - local vPC is down, forwarding via vPC peer-link

vPC domain id          : 10
Peer status             : peer adjacency formed ok
vPC keep-alive status  : peer is alive
Configuration consistency status: success
Per-vlan consistency status : success
Type-2 consistency status : success
vPC role                : primary
Number of vPCs configured : 2
Peer Gateway            : Disabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Enabled

vPC Peer-link status
-----
```

```

id   Port   Status Active vlans
--   -
1    Po1    up     1

vPC status
-----
id   Port   Status Consistency Reason                Active vlans
-----
20   Po20   up     success    success                            1
30   Po30   up     failed     Compatibility check failed 1
                                     for port mode

```

## Viewing a Per-VLAN Consistency Status

To view the per-VLAN consistency or inconsistency status, enter the **show vpc consistency-parameters vlans** command.

This example shows how to display the consistent status of the VLANs on the primary and the secondary switches.

```
switch(config-if)# show vpc brief
```

Legend:

(\*) - local vPC is down, forwarding via vPC peer-link

```

vPC domain id           : 10
Peer status             : peer adjacency formed ok
vPC keep-alive status   : peer is alive
Configuration consistency status: success
Per-vlan consistency status : success
Type-2 consistency status : success
vPC role                : secondary
Number of vPCs configured : 2
Peer Gateway            : Disabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Enabled

```

vPC Peer-link status

```

id   Port   Status Active vlans
--   -
1    Po1    up     1-10

vPC status
-----
id   Port   Status Consistency Reason                Active vlans
-----
20   Po20   up     success    success                            1-10
30   Po30   up     success    success                            1-10

```

Entering **no spanning-tree vlan 5** command triggers the inconsistency on the primary and secondary VLANs:

```
switch(config)# no spanning-tree vlan 5
```

This example shows how to display the per-VLAN consistency status as Failed on the secondary switch:

```
switch(config)# show vpc brief
```

Legend:

(\*) - local vPC is down, forwarding via vPC peer-link

```

vPC domain id           : 10
Peer status             : peer adjacency formed ok
vPC keep-alive status   : peer is alive
Configuration consistency status: success
Per-vlan consistency status : failed
Type-2 consistency status : success
vPC role                : secondary
Number of vPCs configured : 2
Peer Gateway            : Disabled

```

```
Dual-active excluded VLANs      : -
Graceful Consistency Check     : Enabled
```

## vPC Peer-link status

```
-----
id   Port   Status Active vlans
-----
1    Po1    up    1-4,6-10
-----
```

## vPC status

```
-----
id   Port       Status Consistency Reason           Active vlans
-----
20   Po20        up    success    success    1-4,6-10
30   Po30        up    success    success    1-4,6-10
-----
```

This example shows how to display the per-VLAN consistency status as Failed on the primary switch:

```
switch(config)# show vpc brief
```

Legend:

(\*) - local vPC is down, forwarding via vPC peer-link

```
vPC domain id          : 10
Peer status            : peer adjacency formed ok
vPC keep-alive status  : peer is alive
Configuration consistency status: success
Per-vlan consistency status : failed
Type-2 consistency status : success
vPC role               : primary
Number of vPCs configured : 2
Peer Gateway          : Disabled
Dual-active excluded VLANs : -
Graceful Consistency Check : Enabled
```

## vPC Peer-link status

```
-----
id   Port   Status Active vlans
-----
1    Po1    up    1-4,6-10
-----
```

## vPC status

```
-----
id   Port       Status Consistency Reason           Active vlans
-----
20   Po20        up    success    success    1-4,6-10
30   Po30        up    success    success    1-4,6-10
-----
```

This example shows the inconsistency as STP Disabled:

```
switch(config)# show vpc consistency-parameters vlans
```

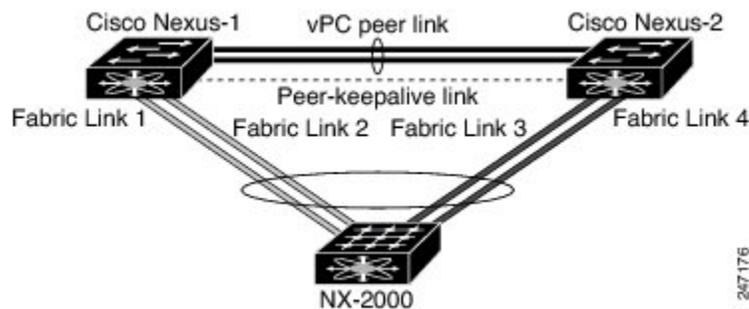
```
Name                                Type Reason Code                Pass Vlans
-----
STP Mode                            1    success                      0-4095
STP Disabled                       1    vPC type-1                   0-4,6-4095
                                     configuration
                                     incompatible - STP is
                                     enabled or disabled on
                                     some or all vlans
STP MST Region Name                  1    success                      0-4095
STP MST Region Revision              1    success                      0-4095
STP MST Region Instance to          1    success                      0-4095
  VLAN Mapping
STP Loopguard                        1    success                      0-4095
STP Bridge Assurance                 1    success                      0-4095
STP Port Type, Edge                 1    success                      0-4095
BPDUFilter, Edge BPDUGuard
STP MST Simulate PVST                1    success                      0-4095
Pass Vlans                            -    -                            0-4,6-4095
```

# vPC Example Configurations

## Dual Homed Fabric Extender vPC Configuration Example

The following example shows how to configure the dual homed Fabric Extender vPC topology using the management VRF to carry the peer-keepalive messages on switch CiscoNexus-1 as shown in following figure:

**Figure 11: vPC Configuration Example**



### Before You Begin

Ensure that the Cisco Nexus 2000 Series Fabric Extender NX-2000-100 is attached and online.

### SUMMARY STEPS

1. Enable vPC and LACP.
2. Create the vPC domain and add the vPC peer-keepalive link.
3. Configure the vPC peer link as a two port Etherchannel.
4. Create a Fabric Extender identifier (for example, "100").
5. Configure the fabric EtherChannel links for the Fabric Extender 100.
6. Configure each host interface port on the Fabric Extender 100 on both Cisco Nexus devices as for all the other steps.
7. Save the configuration.

### DETAILED STEPS

#### Step 1

Enable vPC and LACP.

```
CiscoNexus-1# configure terminal
CiscoNexus-1(config)# feature lACP
CiscoNexus-1(config)# feature vPC
```

**Step 2** Create the vPC domain and add the vPC peer-keepalive link.

```
CiscoNexus-1(config)# vpc domain 1
CiscoNexus-1(config-vpc-domain)# peer-keepalive destination 10.10.10.237
CiscoNexus-1(config-vpc-domain)# exit
```

**Step 3** Configure the vPC peer link as a two port Etherchannel.

```
CiscoNexus-1(config)# interface ethernet 1/1-2
CiscoNexus-1(config-if-range)# switchport mode trunk
CiscoNexus-1(config-if-range)# switchport trunk allowed vlan 20-50
CiscoNexus-1(config-if-range)# switchport trunk native vlan 20
CiscoNexus-1(config-if-range)# channel-group 20 mode active
CiscoNexus-1(config-if-range)# exit
CiscoNexus-1(config)# interface port-channel 20
CiscoNexus-1(config-if)# vpc peer-link
CiscoNexus-1(config-if)# exit
```

**Step 4** Create a Fabric Extender identifier (for example, "100").

```
CiscoNexus-1(config)# fex 100
CiscoNexus-1(config-fex)# pinning max-links 1
CiscoNexus-1(fex)# exit
```

**Step 5** Configure the fabric EtherChannel links for the Fabric Extender 100.

```
CiscoNexus-1(config)# interface ethernet 1/20
CiscoNexus-1(config-if)# channel-group 100
CiscoNexus-1(config-if)# exit
CiscoNexus-1(config)# interface port-channel 100
CiscoNexus-1(config-if)# switchport mode fex-fabric
CiscoNexus-1(config-if)# vpc 100
CiscoNexus-1(config-if)# fex associate 100
CiscoNexus-1(config-if)# exit
```

**Step 6** Configure each host interface port on the Fabric Extender 100 on both Cisco Nexus devices as for all the other steps.

```
CiscoNexus-1(config)# interface ethernet 100/1/1-48
CiscoNexus-1(config-if)# switchport mode access
CiscoNexus-1(config-if)# switchport access vlan 50
CiscoNexus-1(config-if)# no shutdown
CiscoNexus-1(config-if)# exit
```

**Step 7** Save the configuration.

```
CiscoNexus-1(config)# copy running-config startup-config
```

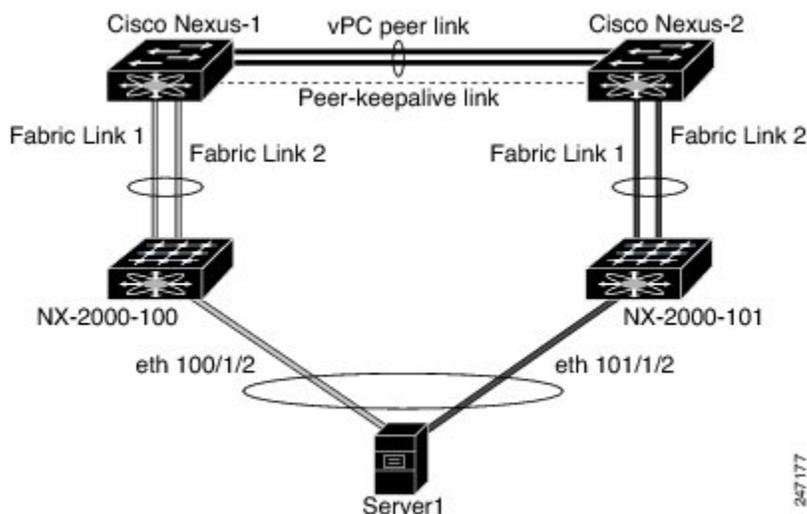
Repeat all the above steps for the CiscoNexus-2 switch.

---

## Single Homed Fabric Extender vPC Configuration Example

The following example shows how to configure the single homed Fabric Extender vPC topology using the default VRF to carry the peer-keepalive messages on switch CiscoNexus-1 as shown in following figure:

**Figure 12: vPC Configuration Example**



### Note

The following example only shows the configuration of CiscoNexus-1 which is connected to the Fabric Extender NX-2000-100. You must repeat these steps on its vPC peer, CiscoNexus-2, which is connected to the Fabric Extender NX-2000-101.

### Before You Begin

Ensure that the Cisco Nexus 2000 Series Fabric Extenders NX-2000-100 and NX-2000-101 are attached and online.

### SUMMARY STEPS

1. Enable vPC and LACP.
2. Enable SVI interfaces, create the VLAN and SVI to be used by the vPC peer-keepalive link.
3. Create the vPC domain and add the vPC peer-keepalive link in the default VRF.
4. Configure the vPC peer link as a two port Etherchannel.
5. Configure the Fabric Extender NX-2000-100.
6. Configure the fabric EtherChannel links for the Fabric Extender NX-2000-100.
7. Configure a vPC server port on on the Fabric Extender NX-2000-100.
8. Save the configuration.

## DETAILED STEPS

### Step 1 Enable vPC and LACP.

```
CiscoNexus-1# configure terminal
CiscoNexus-1(config)# feature lACP
CiscoNexus-1(config)# feature vpc
```

### Step 2 Enable SVI interfaces, create the VLAN and SVI to be used by the vPC peer-keepalive link.

```
CiscoNexus-1(config)# feature interface-vlan
CiscoNexus-1(config)# vlan 900
CiscoNexus-1(config-vlan)# int vlan 900
CiscoNexus-1(config-if)# ip address 10.10.10.236 255.255.255.0
CiscoNexus-1(config-if)# no shutdown
CiscoNexus-1(config-if)# exit
```

### Step 3 Create the vPC domain and add the vPC peer-keepalive link in the default VRF.

```
CiscoNexus-1(config)# vpc domain 30
CiscoNexus-1(config-vpc-domain)# peer-keepalive destination 10.10.10.237 source 10.10.10.236 vrf
default
CiscoNexus-1(config-vpc-domain)# exit
```

**Note** VLAN 900 must **not** be trunked across the vPC peer-link because it carries the vPC peer-keepalive messages. There must be an alternative path between switches CiscoNexus-1 and CiscoNexus-2 for the vPC peer-keepalive messages.

### Step 4 Configure the vPC peer link as a two port Etherchannel.

```
CiscoNexus-1(config)# interface ethernet 1/1-2
CiscoNexus-1(config-if-range)# switchport mode trunk
CiscoNexus-1(config-if-range)# switchport trunk allowed vlan 20-50
CiscoNexus-1(config-if-range)# switchport trunk native vlan 20
CiscoNexus-1(config-if-range)# channel-group 30 mode active
CiscoNexus-1(config-if-range)# exit
CiscoNexus-1(config)# interface port-channel 30
CiscoNexus-1(config-if)# vpc peer-link
CiscoNexus-1(config-if)# exit
```

### Step 5 Configure the Fabric Extender NX-2000-100.

```
CiscoNexus-1(config)# fex 100
CiscoNexus-1(config-fex)# pinning max-links 1
CiscoNexus-1(fex)# exit
```

### Step 6 Configure the fabric EtherChannel links for the Fabric Extender NX-2000-100.

```
CiscoNexus-1(config)# interface ethernet 1/20-21
CiscoNexus-1(config-if)# channel-group 100
CiscoNexus-1(config-if)# exit
CiscoNexus-1(config)# interface port-channel 100
CiscoNexus-1(config-if)# switchport mode fex-fabric
CiscoNexus-1(config-if)# fex associate 100
CiscoNexus-1(config-if)# exit
```

**Step 7** Configure a vPC server port on on the Fabric Extender NX-2000-100.

```
CiscoNexus-1(config-if)# interface ethernet 100/1/1
CiscoNexus-1(config-if)# switchport mode trunk
CiscoNexus-1(config-if)# switchport trunk native vlan 100
CiscoNexus-1(config-if)# switchport trunk allowed vlan 100-105
CiscoNexus-1(config-if)# channel-group 600
CiscoNexus-1(config-if)# no shutdown
CiscoNexus-1(config-if)# exit
CiscoNexus-1(config)# interface port-channel 600
CiscoNexus-1(config-if)# vpc 600
CiscoNexus-1(config-if)# no shutdown
CiscoNexus-1(config-if)# exit
```

**Step 8** Save the configuration.

```
CiscoNexus-1(config)# copy running-config startup-config
```

## vPC Default Settings

The following table lists the default settings for vPC parameters.

**Table 9: Default vPC Parameters**

Parameters	Default
vPC system priority	32667
vPC peer-keepalive message	Disabled
vPC peer-keepalive interval	1 second
vPC peer-keepalive timeout	5 seconds
vPC peer-keepalive UDP port	3200





## Configuring Enhanced Virtual Port Channels

---

This chapter contains the following sections:

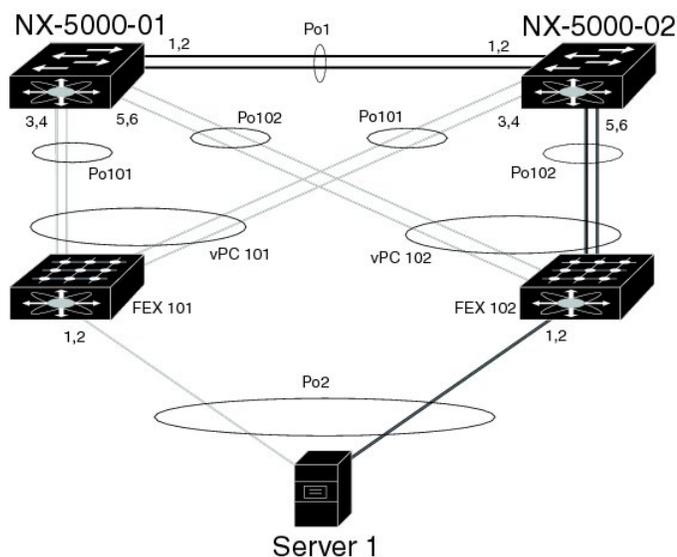
- [Information About Enhanced vPCs, page 150](#)
- [Licensing Requirements for Enhanced vPC, page 152](#)
- [Configuring Enhanced vPCs, page 152](#)
- [Verifying Enhanced vPCs, page 153](#)
- [Enhanced vPC Example Configuration, page 157](#)

# Information About Enhanced vPCs

## Enhanced Virtual Port Channels Overview

The virtual port channel (vPC) feature allows the dual homed connection of a host to two fabric extenders (FEXs) or a dual homed connection of a FEX to two switches. The enhanced vPC feature, or two-layer vPC, allows both dual homing topologies to be combined simultaneously, as shown in the following figure:

**Figure 13: Dual Homing Topology**



With enhanced vPCs, all available paths from the hosts to the FEXs and from the FEXs to the switches are active and carry Ethernet traffic, maximizing the available bandwidth and providing redundancy at both levels. For information about vPCs, see [Configuring Virtual Port Channels](#), on page 101.

## Supported Platforms and Topologies

### Supported Platforms

Enhanced vPC is supported on Cisco Nexus devices.

Any Cisco Nexus Fabric Extender can be used with Enhanced vPC.

Enhanced vPC is compatible with Layer 3 features on the switch.

### Supported and Unsupported Topologies

Enhanced vPC supports the following topologies:

- A single homed server connected to a single FEX

- A dual homed server connected by a port channel to a single FEX
- A dual homed server connected by a port channel to a pair of FEXs

This topology allows connection to any two FEXs that are connected to the same pair of switches in a vPC domain. Static port channel and Link Aggregation Control Protocol (LACP)-based port channel are supported.

- A dual homed server connected by Fibre Channel over Ethernet (FCoE) and port channel to a pair of FEXs
- A dual homed server connected by active/standby NIC teaming to a pair of FEXs

Enhanced vPC does not support the following topologies:

- A dual homed server connected to a pair of FEXs that connect to a single switch  
Although this topology becomes a functioning system when one switch has failed, it is not recommended in normal operation.
- A multi-homed server connected by a port channel to more than two FEXs  
This topology results in increased complexity with little benefit.
- You cannot have a link for non-vPC traffic in parallel with a vPC topology. This can cause errors with the traffic forwarding logic resulting in duplicate or missed packets.

## Enhanced vPC Scalability

The scalability of enhanced vPC is similar to that of the dual homed FEX topology.

Each Cisco Nexus device supports up to 24 FEXs with Layer 2 configuration or Layer 3 configuration. In a dual homed FEX topology, such as that in enhanced vPC, each FEX is managed by two switches, so the pair together can support 24 FEXs.

## Enhanced vPC Failure Response

The enhanced vPC topology provides a high level of resilience to the failure of system components and links as described in the following scenarios:

- Failure of One or More Port Channel Member Links  
When one member link of a port channel fails, the traffic flow is moved to the remaining port channel member links. If all member links of a port channel fail, the traffic flow is redirected to the remaining port channel of the vPC.
- Failure of One FEX  
When one FEX fails, the traffic flow from all dual homed hosts is moved to the remaining FEX.
- Failure of One Switch  
When one switch fails, the traffic flow from all dual homed FEXs is moved to the remaining switch. Traffic from the hosts is unaffected.
- Failure of Both Uplinks from a Single FEX

When both uplinks from one FEX fails, the FEX shuts down its host ports, and the traffic flow from all dual homed hosts is moved to the other FEX.

- Failure of the vPC Peer Link

When the vPC secondary switch detects the failure of the peer link, it checks the status of the primary switch by the peer-keepalive link. If the primary switch is unresponsive, the secondary switch maintains all traffic flows as before. If the primary switch is active, the secondary switch shuts down its interfaces to the FEXs, and the traffic flow from all dual homed FEXs is moved to the primary switch. Ethernet traffic from the hosts is unaffected in either case.

If the secondary switch carries FCoE traffic and shuts down its interfaces to the FEXs, it also shuts down all virtual Fibre Channel (vFC) interfaces that are bound to the FEX host ports. In this case, the hosts must use multipathing to move SAN traffic to the remaining vFC interface.

- Failure of the vPC Peer-Keepalive Link

A failure of the vPC peer-keepalive link by itself does not affect the traffic flow.

## Licensing Requirements for Enhanced vPC

The following table shows the licensing requirements for this feature:

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

## Configuring Enhanced vPCs

### Overview of Configuration Steps for Enhanced vPC

An enhanced vPC configuration consists of a combination of two standard vPC configurations: the dual homed connection of a host to two FEXs and the dual homed connection of a FEX to two switches. The required configuration tasks are listed here, but the detailed procedures for those two standard configurations are presented in the "Configuring Virtual Port Channels" chapter of this document.

To configure enhanced vPC, perform the following steps. Unless noted otherwise, the procedures for each step are given in [Configuring Virtual Port Channels](#), on page 101.

**Note**

- In procedures where the configuration must be repeated on both switches, the configuration synchronization (config-sync) feature allows you to configure one switch and have the configuration automatically synchronized to the peer switch. For more information about configuration synchronization, see the *Operations Guide* for your device.
- You cannot configure non-vPC interfaces across host ports on two different FEXs.

- 
- Step 1** Enable the vPC and LACP features on each switch.
- Step 2** Create required VLANs on each switch.
- Step 3** Assign a vPC domain ID and configure the vPC peer-keepalive link on each switch.
- Step 4** Configure the vPC peer link on each switch.
- Step 5** Configure port channels from the first FEX to each switch.
- Step 6** Configure port channels from the second FEX to each switch.
- Step 7** If the enhanced vPC must accommodate FCoE traffic, associate the first FEX to one switch, and then associate the second FEX to the other switch.  
See “Configuring FCoE over Enhanced vPC” in the *Fibre Channel over Ethernet Configuration Guide* for your device.
- Step 8** Configure a host port channel on each FEX.
- 

## Verifying Enhanced vPCs

### Verifying the Enhanced vPC Configuration

Before bringing up a vPC, the two peer switches in the same vPC domain exchange configuration information to verify that both switches have compatible configurations for a vPC topology. Depending on the severity of the impact of possible mismatched configurations, some configuration parameters are considered as Type 1 consistency check parameters while others are considered as Type 2.

When a mismatch in Type 1 parameters is found, both peer switches suspend VLANs on the vPC ports. When a mismatch in Type 2 parameters is found, a warning syslog message is generated, but the vPC remains up and running.

**Note**

Enhanced vPCs do not support the graceful consistency check.

For enhanced vPCs, the consistency verification for global configuration parameters is the same as for a dual homed FEX topology, and is described in the documentation for dual homed FEX. In addition to the global consistency verification, enhanced vPCs require interface level verification using tasks described in this section.

Use the following commands to verify the enhanced vPC configuration and consistency:

Command	Purpose
switch# <b>show feature</b>	Displays whether vPC is enabled.
switch# <b>show running-config vpc</b>	Displays running configuration information for vPCs.
switch# <b>show vpc brief</b>	Displays brief information on the vPCs.
switch(config)# <b>show vpc consistency-parameters global</b>	Displays the status of global vPC parameters that must be consistent across all vPC interfaces.
switch(config)# <b>show vpc consistency-parameters interface port-channel <i>channel-number</i></b>	Displays the status of specific port channels that must be consistent across vPC devices.

For detailed information about the fields in the output of these commands, see the command reference for your device.

## Verifying the Consistency of Port Channel Numbers

For enhanced vPCs, both switches must use the same port channel number for the dual homed connection to a FEX. If different port channel numbers are used, the port channel and its member ports are suspended on both switches.

### SUMMARY STEPS

1. **show running-config interface *type/slot* [, *type/slot* [, ...]]**
2. **show interface *type/slot***

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>show running-config interface <i>type/slot</i> [, <i>type/slot</i> [, ...]]</b>  <b>Example:</b> switch-1# show running-config interface Ethernet110/1/1, Ethernet111/1/1	Displays the configuration of the specified list of port channel member ports.  Enter this command on both peer switches and compare the reported <b>channel-group</b> numbers to verify that they match between switches.
<b>Step 2</b>	<b>show interface <i>type/slot</i></b>  <b>Example:</b> switch-1# show interface Ethernet110/1/1	Displays the status and configuration of the specified port channel member port.  Enter this command on both peer switches and verify the status of the ports.

This example shows how to verify the consistency of the port channel numbering between the two switches. In this example, the port channel numbering is inconsistent and the member ports are suspended:

```
switch-1# show running-config interface Ethernet110/1/1, Ethernet111/1/1

!Command: show running-config interface Ethernet110/1/1, Ethernet111/1/1
!Time: Sun Aug 28 03:38:23 2011

version 5.1(3)N1(1)

interface Ethernet110/1/1
channel-group 102

interface Ethernet111/1/1
channel-group 102

switch-2# show running-config interface Ethernet110/1/1, Ethernet111/1/1

!Command: show running-config interface Ethernet110/1/1, Ethernet111/1/1
!Time: Sun Aug 28 03:38:23 2011

version 5.1(3)N1(1)

interface Ethernet110/1/1
channel-group 101

interface Ethernet111/1/1
channel-group 101

switch-1# show interface Ethernet110/1/1
Ethernet110/1/1 is down (suspended by vpc)
  Hardware: 100/1000 Ethernet, address: 7081.0500.2402 (bia 7081.0500.2402)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec
  [...]

switch-2# show interface Ethernet110/1/1
Ethernet110/1/1 is down (suspended by vpc)
  Hardware: 100/1000 Ethernet, address: 7081.0500.2402 (bia 7081.0500.2402)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec
  [...]
```

## Verifying Common Port Channel Members

The port channel from a FEX to the switch pair is up and operational when there is at least one common port channel member between the two switches. Any FEX interfaces that are assigned to the port channel only on one switch will be suspended.

### SUMMARY STEPS

1. show port-channel summary
2. (Optional) show interface *type/slot*

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>show port-channel summary</b>  <b>Example:</b> switch-1# show port-channel summary	Displays a summary of the port channel interfaces.

	Command or Action	Purpose
Step 2	<b>show interface</b> <i>type/slot</i>  <b>Example:</b> switch-1# show interface ethernet 111/1/3	(Optional) Displays the status and configuration of the specified interface.

This example shows how to verify the common member ports of the vPC. In this example, the vPC is configured with one port channel member that is not common to both switches. That member port is shown as shut down, and further investigation shows that the member is suspended by the vPC. In this part of the session, the port channel is configured on each switch, with an extra port on the first switch:

```
switch-1(config)# interface ethernet 110/1/3, ethernet 111/1/3
switch-1(config-if)# channel-group 101
switch-1(config-if)# interface port-channel 101
switch-1(config-if)# switchport access vlan 20

switch-2(config)# interface ethernet 110/1/3
switch-2(config-if)# channel-group 101
switch-2(config-if)# interface port-channel 101
switch-2(config-if)# switchport access vlan 20
```

In this part of the session, the extra port is shown to be in the down state, and a display of the port details shows that the port is suspended by the vPC:

```
switch-1# show port-channel summary
Flags: D - Down          P - Up in port-channel (members)
       I - Individual    H - Hot-standby (LACP only)
       s - Suspended     r - Module-removed
       S - Switched      R - Routed
       U - Up (port-channel)
       M - Not in use. Min-links not met

-----
Group Port-      Type      Protocol  Member Ports
Channel
-----
1      Po1(SU)     Eth       LACP      Eth1/1(P)   Eth1/2(P)
[...]
101    Po101(SU)   Eth       NONE      Eth110/1/3(P)  Eth111/1/3(D)
```

```
switch-1# show interface ethernet 111/1/3
Ethernet111/1/3 is down (suspended by vpc)
  Hardware: 100/1000 Ethernet, address: 7081.0500.2582 (bia 7081.0500.2582)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec
  reliability 255/255, txload 1/255, rxload 1/255
```

## Verifying Interface Level Consistency for Enhanced vPCs

For enhanced vPCs, you must ensure consistency of the port mode and the shared VLAN in the port channel interface configuration.

### SUMMARY STEPS

1. **show vpc consistency-parameters port-channel** *channel-number*

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>show vpc consistency-parameters port-channel</b> <i>channel-number</i>  <b>Example:</b> <pre>switch# show vpc consistency-parameters interface port-channel 101 switch(config)#</pre>	For the specified port channel, displays the status information that must be consistent across vPC devices.

This example shows how to display a comparison of the interface configuration across two peers for a vPC. In this case, VLAN 10 is allowed on both peers, but the port mode is mismatched, causing the VLAN to be suspended.

```
switch-1# show vpc consistency-parameters interface port-channel 101
```

Legend:

Type 1 : vPC will be suspended in case of mismatch

Name	Type	Local Value	Peer Value
mode	1	on	on
Speed	1	1000 Mb/s	1000 Mb/s
Duplex	1	full	full
Port Mode	1	<b>access</b>	<b>trunk</b>
MTU	1	1500	1500
Admin port mode	1		
Shut Lan	1	No	No
vPC+ Switch-id	1	3000	3000
Allowed VLANs	-	<b>10</b>	<b>1-57</b> , 61-3967, 4048-4093
Local suspended VLANs	-	10	-

## Enhanced vPC Example Configuration

The following example shows the complete configuration procedure using the topology of the enhanced vPC figure in this chapter. In the topology figure, the number pairs beside each port channel link represent the interface port numbers. For example, the switch link labeled with the numbers "3,4" represents interfaces eth1/3 and eth1/4 on the switch.



### Note

In procedures where the configuration must be repeated on both switches, the configuration synchronization (config-sync) feature allows you to configure one switch and have the configuration automatically synchronized to the peer switch. For more information about configuration synchronization, see the operations guide for your device.

### Before You Begin

Ensure that the Cisco Nexus Fabric Extenders FEX101 and FEX102 are attached and online.

**SUMMARY STEPS**

1. Enable the vPC and LACP features on each switch.
2. Create required VLANs on each switch.
3. Assign a vPC domain ID and configure the vPC peer-keepalive link on each switch.
4. Configure the vPC peer link on each switch.
5. Configure port channels from the first FEX to each switch.
6. Configure port channels from the second FEX to each switch.
7. Configure a host port channel on each FEX.

**DETAILED STEPS**

**Step 1** Enable the vPC and LACP features on each switch.

**Example:**

```
switch-1(config)# feature vpc
switch-1(config)# feature lacp

switch-2(config)# feature vpc
switch-2(config)# feature lacp
```

**Step 2** Create required VLANs on each switch.

**Example:**

```
switch-1(config)# vlan 10-20

switch-2(config)# vlan 10-20
```

**Step 3** Assign a vPC domain ID and configure the vPC peer-keepalive link on each switch.

**Example:**

```
switch-1(config)# vpc domain 123
switch-1(config-vpc)# peer-keepalive destination 172.25.182.100

switch-2(config)# vpc domain 123
switch-2(config-vpc)# peer-keepalive destination 172.25.182.99
```

**Note** When you configure each switch, use the IP address of the peer switch as the peer-keepalive destination.

**Step 4** Configure the vPC peer link on each switch.

**Example:**

```
switch-1(config)# interface eth1/1-2
switch-1(config-if)# channel-group 1 mode active
switch-1(config-if)# interface Po1
switch-1(config-if)# switchport mode trunk
switch-1(config-if)# switchport trunk allowed vlan 1, 10-20
switch-1(config-if)# vpc peer-link

switch-2(config)# interface eth1/1-2
switch-2(config-if)# channel-group 1 mode active
switch-2(config-if)# interface Po1
switch-2(config-if)# switchport mode trunk
switch-2(config-if)# switchport trunk allowed vlan 1, 10-20
```

```
switch-2(config-if)# vpc peer-link
```

**Step 5** Configure port channels from the first FEX to each switch.

**Example:**

```
switch-1(config)# fex 101
switch-1(config-fex)# interface eth1/3-4
switch-1(config-if)# channel-group 101
switch-1(config-if)# interface po101
switch-1(config-if)# switchport mode fex-fabric
switch-1(config-if)# vpc 101
switch-1(config-if)# fex associate 101
```

```
switch-2(config)# fex 101
switch-2(config-fex)# interface eth1/3-4
switch-2(config-if)# channel-group 101
switch-2(config-if)# interface po101
switch-2(config-if)# switchport mode fex-fabric
switch-2(config-if)# vpc 101
switch-2(config-if)# fex associate 101
```

**Step 6** Configure port channels from the second FEX to each switch.

**Example:**

```
switch-1(config)# fex 102
switch-1(config-fex)# interface eth1/5-6
switch-1(config-if)# channel-group 102
switch-1(config-if)# interface po102
switch-1(config-if)# switchport mode fex-fabric
switch-1(config-if)# vpc 102
switch-1(config-if)# fex associate 102
```

```
switch-2(config)# fex 102
switch-2(config-fex)# interface eth1/5-6
switch-2(config-if)# channel-group 102
switch-2(config-if)# interface po102
switch-2(config-if)# switchport mode fex-fabric
switch-2(config-if)# vpc 102
switch-2(config-if)# fex associate 102
```

**Step 7** Configure a host port channel on each FEX.

**Example:**

```
switch-1(config)# interface eth101/1/1, eth101/1/2
switch-1(config-if)# channel-group 2 mode active
switch-1(config-if)# interface eth102/1/1, eth102/1/2
switch-1(config-if)# channel-group 2 mode active
switch-1(config-if)# int po2
switch-1(config-if)# switchport access vlan 10
```

```
switch-2(config)# interface eth101/1/1, eth101/1/2
switch-2(config-if)# channel-group 2 mode active
switch-2(config-if)# interface eth102/1/1, eth102/1/2
switch-2(config-if)# channel-group 2 mode active
switch-2(config-if)# int po2
switch-2(config-if)# switchport access vlan 10
```





## Configuring Rapid PVST+

This chapter contains the following sections:

- [Information About Rapid PVST+, page 161](#)
- [Configuring Rapid PVST+, page 176](#)
- [Verifying the Rapid PVST+ Configuration, page 186](#)

### Information About Rapid PVST+

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 802.1D 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 in the software. Rapid PVST+ interoperates with switches that run legacy IEEE 802.1D STP.

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



**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.

---

### Understanding STP

#### STP Overview

For an Ethernet network to function properly, only one active path can exist between any two stations.

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 network. LAN ports send and receive STP frames, which are called Bridge Protocol Data Units (BPDUs), at regular intervals. Switches 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 switches might learn end station MAC addresses on multiple LAN ports. These conditions result in a broadcast storm, which creates an unstable network.

STP defines a tree with a root bridge and a loop-free path from the root to all switches in the 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 LAN ports on a switch are part of a loop, the STP port priority and port path cost setting determine which port on the switch is put in the forwarding state and which port is put in the blocking state.

## Understanding How a Topology is Created

All switches in an extended LAN that participate in a spanning tree gather information about other switches in the network by exchanging of 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 interfaces in a backup state; all paths that are not needed to reach the root switch 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 switch identifier Media Access Control (MAC) address of the switch that is associated with each switch
- The path cost to the root that is associated with each interface
- The port identifier that is associated with each interface

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, as well as the root port and designated port for each switched segment.

## Understanding the Bridge ID

Each VLAN on each switch 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.

**Note**

---

In Cisco NX-OS, the extended system ID is always enabled; you cannot disable the extended system ID.

---

### Extended System ID

A 12-bit extended system ID field is part of the bridge ID.

**Figure 14: Bridge ID with Extended System ID**



The switches always use the 12-bit extended system ID.

Combined with the bridge ID, the system ID extension functions as the unique identifier for a VLAN.

**Table 10: 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**

Extended system ID and MAC address reduction is always enabled on the software.

With MAC address reduction enabled on any switch, you should also enable MAC address reduction on all other connected switches 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 switch bridge ID (used by the spanning tree algorithm to determine the identity of the root bridge, the lowest being 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 achieve root bridge ownership because its bridge ID may fall between the values specified by the MAC address reduction feature.

## Understanding BPDUs

Switches transmit bridge protocol data units (BPDUs) throughout the STP instance. Each switch 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 switch that the transmitting switch determines is the root bridge
- The STP path cost to the root
- The bridge ID of the transmitting bridge
- 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 switch transmits a Rapid PVST+ BPDU frame, all switches connected to the VLAN on which the frame is transmitted receive the BPDU. When a switch receives a BPDU, it does not forward the frame but instead uses the information in the frame to calculate a BPDU, and, if the topology changes, initiate a BPDU transmission.

A BPDU exchange results in the following:

- One switch is elected as the root bridge.
- The shortest distance to the root bridge is calculated for each switch based on the path cost.
- A designated bridge for each LAN segment is selected. This is the switch closest to the root bridge through which frames are forwarded to the root.
- A root port is selected. This is the port providing 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 switch with the lowest numerical value of the bridge ID is elected as the root bridge. If all switches are configured with the default priority (32768), the switch 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 switch will be elected as the root bridge. Configuring a lower value increases the probability; a higher value decreases the probability.

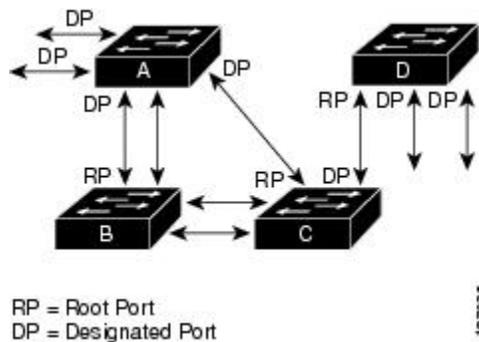
The STP root bridge is the logical center of each spanning tree topology in a network. All paths that are not needed to reach the root bridge from anywhere in the network are placed in STP blocking mode.

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 leading to the root bridge, and to determine the designated port for each segment.

## Creating the Spanning Tree Topology

In the following figure, Switch A is elected as the root bridge because the bridge priority of all the switches 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. By increasing the priority (lowering the numerical value) of the ideal switch so that it becomes the root bridge, you force an STP recalculation to form a new spanning tree topology with the ideal switch as the root.

Figure 15: Spanning Tree Topology



When the spanning tree topology is calculated based on default parameters, the path between 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.

# Understanding Rapid PVST+

## Rapid PVST+ Overview

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 switch.

---

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).




---

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

---

Using Rapid PVST+, STP convergence occurs rapidly. Each designated or root port in the STP sends out a BPDU every 2 seconds by default. On a designated or root 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 root or designated port if it misses three BPDUs or if the maximum age expires. This rapid aging of the protocol information allows quick failure detection. The switch automatically checks the PVID.

Rapid PVST+ provides for rapid recovery of connectivity following the failure of a network device, a switch 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 switch, the edge port immediately transitions to the forwarding state. (This immediate transition was previously a Cisco-proprietary feature named PortFast.) You should only configure on 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 host as edge ports.

---

- Root ports—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 out 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 non-edge 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.



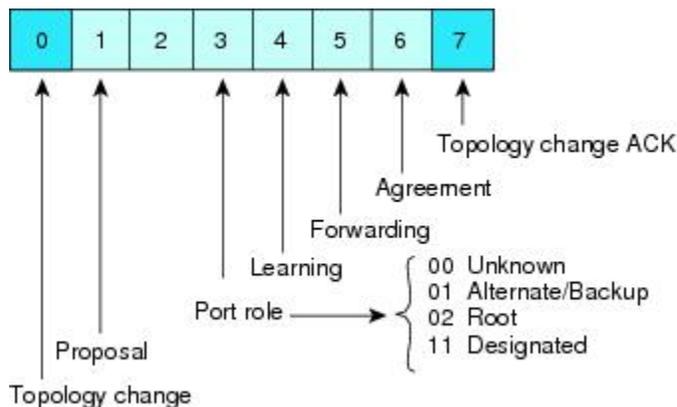
**Note** The TCA flag is used only when the switch is interacting with switches that are running legacy 802.1D STP.

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 role and state of the port that originates the BPDU and the proposal and agreement handshake. The following figure shows the use of the BPDU flags in Rapid PVST+.

**Figure 16: Rapid PVST+ Flag Byte in BPDU**

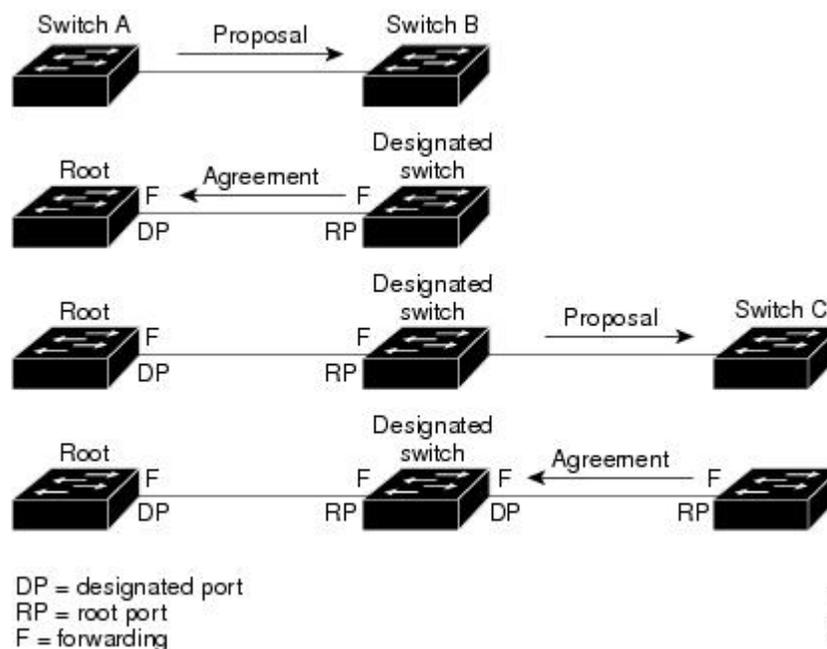


Another important change is that the Rapid PVST+ BPDU is type 2, version 2, which makes it possible for the switch to detect connected legacy (802.1D) bridges. The BPDU for 802.1D is version 0.

## Proposal and Agreement Handshake

As shown in the following 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.

**Figure 17: Proposal and Agreement Handshaking for Rapid Convergence**



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 non-edge 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 non-edge 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 network device joins the active topology. As the network converges, this proposal-agreement handshaking progresses from the root toward the leaves of the spanning tree.

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 non-edge port moves from the blocking to the forwarding state. The handshaking process then proliferates step-by-step throughout the topology.

## Protocol Timers

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

**Table 11: Rapid PVST+ Protocol Timers**

Variable	Description
Hello timer	Determines how often each switch broadcasts BPDUs to other switches. 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 is used as a backup. The default is 15 seconds, and the range is from 4 to 30 seconds.
Maximum age timer	Determines the amount of time protocol information received on an port is stored by the switch. This timer is generally not used by the protocol, but it is used when interoperating with 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 switch with the highest priority (lowest numerical priority value) as the root bridge. Rapid PVST+ then assigns one of these port roles to individual ports:

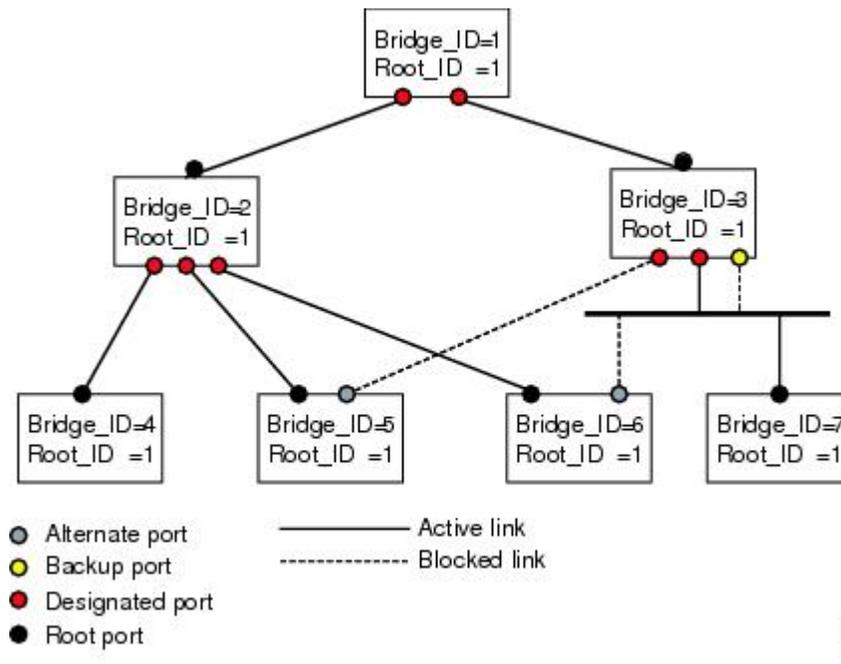
- Root port—Provides the best path (lowest cost) when the switch forwards packets to the root bridge.
- Designated port—Connects to the designated switch, which incurs the lowest path cost when forwarding packets from that LAN to the root bridge. The port through which the designated switch is attached 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 switch 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 switch has two or more connections to a shared LAN segment. A backup port provides another path in the topology to the switch.
- 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.

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 (see the following figure).

**Figure 18: Sample Topology Demonstrating Port Roles**



## Port States

### 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 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 LAN port on a software using Rapid PVST+ or MST exists in one of the following four states:

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

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

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

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

### Blocking State

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

A 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 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 network management messages.

### Learning State

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

A 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 network management messages.

### Forwarding State

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

A 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 network management messages.

### Disabled State

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

A disabled 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

The following table lists the possible operational and Rapid PVST+ states for ports and the corresponding inclusion in the active topology.

**Table 12: 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 switch 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.

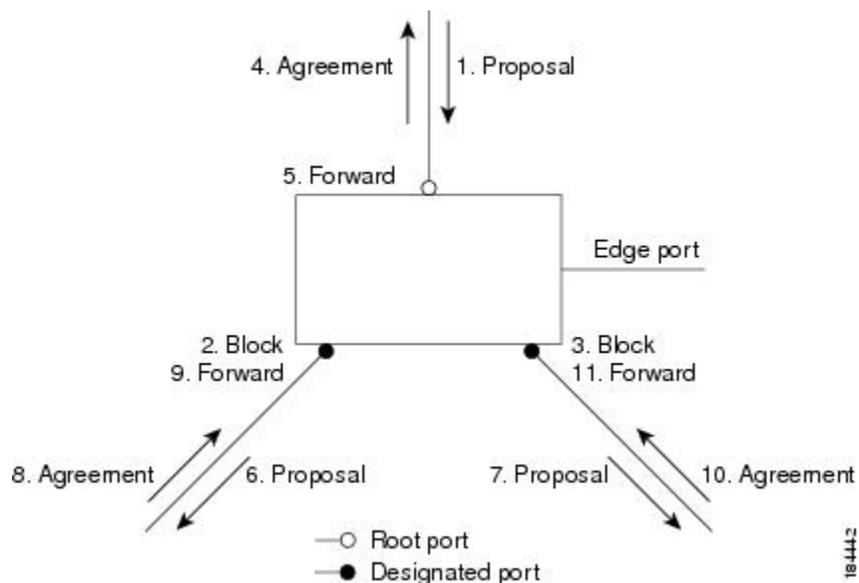
The switch is synchronized with superior root information received on the root port if all other ports are synchronized. An individual port on the switch is synchronized if either of the following applies:

- 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 switch sends an agreement message to the designated switch that corresponds to its root port. When the switches 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. The sequence of events is shown in the following figure.

**Figure 19: Sequence of Events During Rapid Convergence**



### 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 other ports to synchronize.

If the received BPDU is a Rapid PVST+ BPDU with the proposal flag set, the switch 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.

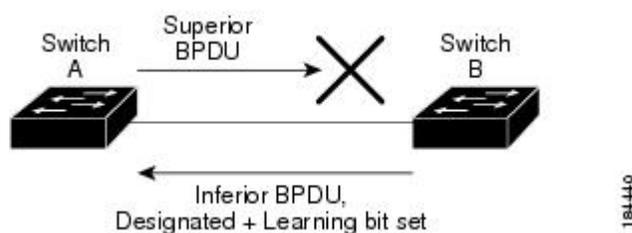
## Spanning-Tree Dispute Mechanism

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.

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.

The following 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 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. The block is shown as an STP dispute.

**Figure 20: Detecting Unidirectional Link Failure**



## 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 switch 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.

The STP port path-cost default value is determined from the media speed and path-cost calculation method of a LAN interface. If a loop occurs, STP considers the port cost when selecting a LAN interface to put into the forwarding state.

**Table 13: Default Port Cost**

Bandwidth	Short Path-Cost Method of Port Cost	Long Path-Cost Method of Port Cost
10 Mbps	100	2,000,000

Bandwidth	Short Path-Cost Method of Port Cost	Long Path-Cost Method of Port Cost
100 Mbps	19	200,000
1 Gigabit Ethernet	4	20,000
10 Gigabit Ethernet	2	2,000

You can assign 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 loop 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 software 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

In a network of Cisco switches connected through 802.1Q trunks, the switches maintain one instance of STP for each VLAN allowed on the trunks. However, non-Cisco 802.1Q switches maintain only one instance of STP for all VLANs allowed on the trunks.

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

## Rapid PVST+ Interoperation with Legacy 802.1D STP

Rapid PVST+ can interoperate with switches that are running the legacy 802.1D protocol. The switch 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 switch sends an agreement message after all of the other ports are synchronized. If the BPDU is an 802.1D

BPDU version 0, the switch 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 switch interoperates with legacy 802.1D switches as follows:

- **Notification**—Unlike 802.1D BPDUs, 802.1w does not use TCN BPDUs. However, for interoperability with 802.1D switches, Cisco NX-OS processes and generates TCN BPDUs.
- **Acknowledgement**—When an 802.1w switch receives a TCN message on a designated port from an 802.1D switch, 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 switch 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 switches. The 802.1w BPDUs do not have the TCA bit set.

- **Protocol migration**—For backward compatibility with 802.1D switches, 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 switch processes all BPDUs received on that port and ignores the protocol type.

If the switch receives an 802.1D BPDU after the port migration-delay timer has expired, it assumes that it is connected to an 802.1D switch and starts using only 802.1D BPDUs. However, if the 802.1w switch 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 switches to renegotiate the protocol, you must restart 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.

## Configuring Rapid PVST+

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

You enable Rapid PVST+ on a per-VLAN basis. The software 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+

Once you enable Rapid PVST+ on the switch, you must enable Rapid PVST+ on the specified VLANs.

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



**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.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mode rapid-pvst**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	switch(config)# <b>spanning-tree mode rapid-pvst</b>	Enables Rapid PVST+ on the switch. Rapid PVST+ is the default spanning tree mode.  <b>Note</b> Changing the spanning tree mode disrupts traffic because all spanning tree instances are stopped for the previous mode and started for the new mode.

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

```
switch# configure terminal
switch(config)# spanning-tree mode rapid-pvst
```



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

## Enabling Rapid PVST+ per VLAN

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. switch# **configure terminal**
2. switch(config)# **spanning-tree vlan-range**
3. (Optional) switch(config)# **no spanning-tree vlan-range**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>spanning-tree</b> <i>vlan-range</i>	Enables Rapid PVST+ (default STP) on a per VLAN basis. The <i>vlan-range</i> value can be 2 through 4094 (except reserved VLAN values).
<b>Step 3</b>	switch(config)# <b>no</b> <b>spanning-tree</b> <i>vlan-range</i>	(Optional) Disables Rapid PVST+ on the specified VLAN.  <b>Caution</b> 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 of the switches and bridges in a VLAN and leave it enabled on other switches and bridges. 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.  Do not disable spanning tree in a VLAN without ensuring that there are no physical loops present in the VLAN. Spanning tree serves as a safeguard against misconfigurations and cabling errors.

This example shows how to enable STP on a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5
```

## Configuring the Root Bridge ID

The software maintains a separate instance of STP for each active VLAN in Rapid PVST+. For each VLAN, the switch 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\_ID* root** command, the switch checks the bridge priority of the current root bridges for each VLAN. The switch sets the bridge priority for the specified VLANs to 24576 if this value will cause the switch to become the root for the specified VLANs. If any root bridge for the specified VLANs has a bridge priority lower than 24576, the switch sets the bridge priority for the specified VLANs to 4096 less than the lowest bridge priority.

**Note**

The **spanning-tree vlan *vlan\_ID* root** command fails if the value required to be the root bridge is less than 1.

**Caution**

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

Enter the **diameter** keyword to specify the network diameter (that is, the maximum number of bridge hops between any two end stations in the 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.



**Note** With the switch 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** configuration commands.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree vlan** *vlan-range* **root primary** [**diameter** *dia* [**hello-time** *hello-time*]]

## DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	switch(config)# <b>spanning-tree vlan</b> <i>vlan-range</i> <b>root primary</b> [ <b>diameter</b> <i>dia</i> [ <b>hello-time</b> <i>hello-time</i> ]]	Configures a software switch as the primary root bridge. The <i>vlan-range</i> value can be 2 through 4094 (except reserved VLAN values.) The <i>dia</i> default is 7. The <i>hello-time</i> can be from 1 to 10 seconds, and the default value is 2 seconds.

This example shows how to configure the switch as the root bridge for a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5 root primary diameter 4
```

## Configuring a Secondary Root Bridge

When you configure a software switch as the secondary root, the STP bridge priority is modified from the default value (32768) so that the switch is likely to become the root bridge for the specified VLANs if the primary root bridge fails (assuming the other switches in the network use the default bridge priority of 32768). STP sets the bridge priority to 28672.

Enter the **diameter** keyword to specify the network diameter (that is, the maximum number of bridge hops between any two end stations in the 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 configure more than one switch 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 switch 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. switch# **configure terminal**
2. switch(config)# **spanning-tree vlan *vlan-range* root secondary [diameter *dia* [hello-time *hello-time*]]**

**DETAILED STEPS**

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>spanning-tree vlan <i>vlan-range</i> root secondary [diameter <i>dia</i> [hello-time <i>hello-time</i>]]</b>	Configures a software switch as the secondary root bridge. The <i>vlan-range</i> value can be 2 through 4094 (except reserved VLAN values). The <i>dia</i> default is 7. The <i>hello-time</i> can be from 1 to 10 seconds, and the default value is 2 seconds.

This example shows how to configure the switch as the secondary root bridge for a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5 root secondary diameter 4
```

## Configuring the Rapid PVST+ Port Priority

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 software uses the port priority value when the LAN port is configured as an access port and uses VLAN port priority values when the LAN port is configured as a trunk port.

**SUMMARY STEPS**

1. switch# **configure terminal**
2. switch(config)# **interface *type slot/port***
3. switch(config-if)# **spanning-tree [vlan *vlan-list*] port-priority *priority***

## DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	switch(config)# <b>interface</b> <i>type slot/port</i>	Specifies the interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# <b>spanning-tree</b> [ <b>vlan</b> <i>vlan-list</i> ] <b>port-priority</b> <i>priority</i>	Configures the port priority for the LAN interface. The <i>priority</i> value can be from 0 to 224. The lower the value indicates the higher the priority. The priority values are 0, 32, 64, 96, 128, 160, 192, and 224. All other values are rejected. The default value is 128.

This example shows how to configure the access port priority of an Ethernet interface:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree port-priority 160
```

You can only apply this command to a physical Ethernet interface.

## Configuring the Rapid PVST+ Path-Cost Method and Port Cost

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


**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 submenu. The default path-cost method is short.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree pathcost method** {**long** | **short**}
3. switch(config)# **interface** *type slot/port*
4. switch(config-if)# **spanning-tree** [**vlan** *vlan-id*] **cost** [*value* | **auto**]

## DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	switch(config)# <b>spanning-tree pathcost method</b> { <b>long</b>   <b>short</b> }	Selects the method used for Rapid PVST+ path-cost calculations. The default method is the short method.

	Command or Action	Purpose
<b>Step 3</b>	switch(config)# <b>interface</b> <i>type slot/port</i>	Specifies the interface to configure, and enters interface configuration mode.
<b>Step 4</b>	switch(config-if)# <b>spanning-tree</b> [ <b>vlan</b> <i>vlan-id</i> ] <b>cost</b> [ <i>value</i>   <b>auto</b> ]	Configures the port cost for the LAN interface. The cost value, depending on the path-cost calculation method, can be as follows: <ul style="list-style-type: none"> <li>• short—1 to 65535</li> <li>• long—1 to 200000000</li> </ul> <p><b>Note</b> You configure this parameter per interface on access ports and per VLAN on trunk ports. The default is <b>auto</b>, which sets the port cost on both the path-cost calculation method and the media speed.</p>

This example shows how to configure the access port cost of an Ethernet interface:

```
switch# configure terminal
switch (config)# spanning-tree pathcost method long
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree cost 1000
```

You can only apply this command to a physical Ethernet interface.

## Configuring the Rapid PVST+ Bridge Priority of a VLAN

You can configure the Rapid PVST+ bridge priority of a VLAN.



### Note

Be careful when using this configuration. For most situations, we recommend that you configure the primary root and secondary root to modify the bridge priority.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree vlan** *vlan-range* **priority** *value*

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>spanning-tree vlan</b> <i>vlan-range</i> <b>priority</b> <i>value</i>	Configures the bridge priority of a VLAN. Valid values are 0, 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768, 36864, 40960, 45056,

	Command or Action	Purpose
		49152, 53248, 57344, and 61440. All other values are rejected. The default value is 32768.

This example shows how to configure the bridge priority of a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5 priority 8192
```

## Configuring the Rapid PVST+ Hello Time for a VLAN

You can configure the Rapid PVST+ hello time for a VLAN.



### Note

Be careful when using this configuration. For most situations, we recommend that you configure the primary root and secondary root to modify the hello time.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree vlan *vlan-range* hello-time *hello-time***

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>spanning-tree vlan <i>vlan-range</i> hello-time <i>hello-time</i></b>	Configures the hello time of a VLAN. The hello time value can be from 1 to 10 seconds. The default is 2 seconds.

This example shows how to configure the hello time for a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5 hello-time 7
```

## Configuring the Rapid PVST+ Forward Delay Time for a VLAN

You can configure the forward delay time per VLAN when using Rapid PVST+.

**SUMMARY STEPS**

1. switch# **configure terminal**
2. switch(config)# **spanning-tree vlan *vlan-range* forward-time *forward-time***

**DETAILED STEPS**

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>spanning-tree vlan <i>vlan-range</i> forward-time <i>forward-time</i></b>	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.

This example shows how to configure the forward delay time for a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5 forward-time 21
```

**Configuring the Rapid PVST+ Maximum Age Time for a VLAN**

You can configure the maximum age time per VLAN when using Rapid PVST+.

**SUMMARY STEPS**

1. switch# **configure terminal**
2. switch(config)# **spanning-tree vlan *vlan-range* max-age *max-age***

**DETAILED STEPS**

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters configuration mode.
<b>Step 2</b>	switch(config)# <b>spanning-tree vlan <i>vlan-range</i> max-age <i>max-age</i></b>	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.

This example shows how to configure the maximum aging time for a VLAN:

```
switch# configure terminal
switch(config)# spanning-tree vlan 5 max-age 36
```

## Specifying the Link Type

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 switch, you can override the default setting on the link type and enable rapid transitions.

If you set the link to shared, STP moves back to 802.1D.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot/port*
3. switch(config-if)# **spanning-tree link-type** {**auto** | **point-to-point** | **shared**}

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface</b> <i>type slot/port</i>	Specifies the interface to configure, and enters the interface configuration mode.
<b>Step 3</b>	switch(config-if)# <b>spanning-tree link-type</b> { <b>auto</b>   <b>point-to-point</b>   <b>shared</b> }	Configures the link type to be either a point-to-point link or shared link. The system reads the default value from the switch connection, as follows: half duplex links are shared and full-duplex links are point-to-point. If the link type is shared, the STP reverts to 802.1D. The default is auto, which sets the link type based on the duplex setting of the interface.

This example shows how to configure the link type as a point-to-point link:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree link-type point-to-point
```

You can only apply this command to a physical Ethernet interface.

## Restarting the Protocol

A bridge running 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 switch has been removed from the link unless the legacy switch is the designated switch. You can restart the protocol negotiation (force the renegotiation with neighboring switches) on the entire switch or on specified interfaces.

Command	Purpose
switch# <b>clear spanning-tree detected-protocol</b> [ <b>interface interface</b> [ <i>interface-num</i>   <i>port-channel</i> ]]	Restarts Rapid PVST+ on all interfaces on the switch or specified interfaces.

This example shows how to restart Rapid PVST+ on an Ethernet interface:

```
switch# clear spanning-tree detected-protocol interface ethernet 1/8
```

## Verifying the Rapid PVST+ Configuration

Use the following commands to display Rapid PVST+ configuration information.

Command	Purpose
<b>show running-config spanning-tree</b> [all]	Displays the current spanning tree configuration.
<b>show spanning-tree</b> [options]	Displays selected detailed information for the current spanning tree configuration.

This example shows how to display spanning tree status:

```
switch# show spanning-tree brief

VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    32768
             Address    001c.b05a.5447
             Cost        2
             Port        131 (Ethernet1/3)
             Hello Time  2 sec    Max Age 20 sec    Forward Delay 15 sec
  Bridge ID  Priority    32769 (priority 32768 sys-id-ext 1)
             Address    000d.ec6d.7841
             Hello Time  2 sec    Max Age 20 sec    Forward Delay 15 sec
Interface   Role Sts Cost          Prio.Nbr Type
-----
Eth1/3      Root FWD 2             128.131 P2p Peer (STP)
veth1/1     Desg FWD 2             128.129 Edge P2p
```



# Configuring Multiple Spanning Tree

This chapter contains the following sections:

- [Information About MST, page 187](#)
- [Configuring MST, page 195](#)
- [Verifying the MST Configuration, page 213](#)

## Information About MST

### MST Overview



**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.

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 provides rapid convergence through explicit handshaking as 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 while you are using MST. 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+)  
IEEE 802.1w defined the Rapid Spanning Tree Protocol (RSTP) and was incorporated into IEEE 802.1D.
- IEEE 802.1s defined MST and was incorporated into IEEE 802.1Q.

**Note**

You must enable MST; Rapid PVST+ is the default spanning tree mode.

## MST Regions

To allow switches to participate in MST instances, you must consistently configure the switches with the same MST configuration information.

A collection of interconnected switches 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 switch belongs. The configuration includes the name of the region, the revision number, and the MST VLAN-to-instance assignment map.

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 region can support up to 65 MST instances (MSTIs). 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.

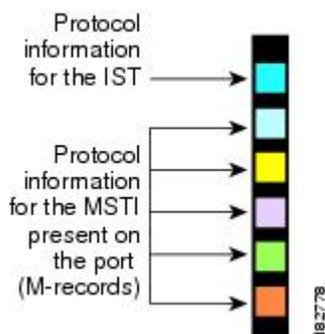
**Note**

We recommend that you do not partition the network into a large number of regions.

## MST BPDUs

Each region has only one MST BPDU, and that BPDU carries an M-record for each MSTI within the region (see the following figure). 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 MSTIs is significantly reduced.

**Figure 21: MST BPDU with M-Records for MSTIs**



## MST Configuration Information

The MST configuration that must be identical on all switches within a single MST region is configured by the user.

You can configure the following three parameters of the MST configuration:

- 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.

- MST configuration table—4096-element table that associates each of the potential 4094 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.

**Caution**

When you change the VLAN-to-MSTI mapping, the system restarts 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.

- 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 switches 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 MST switches 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 switches at the boundary of the region as the CIST regional root.

When an MST switch 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 switch also initializes all of its MSTIs and claims to be the root for all of them. If the switch receives superior MST 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 switches 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 switches in the MST region must agree on the same CIST regional root. Any two switches in the region will only synchronize their port roles for an MSTI if they converge to a common CIST regional root.

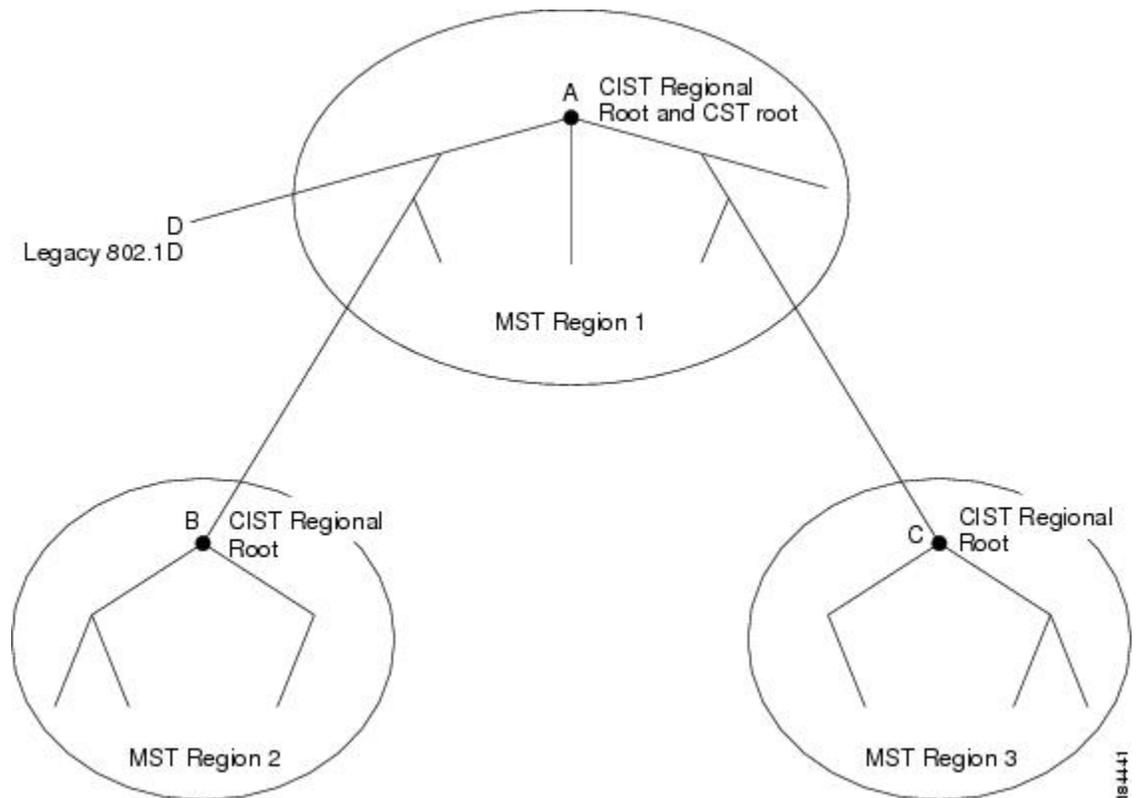
## Spanning Tree Operations Between MST Regions

If you have multiple regions or 802.1w 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 switches 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 switches 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 switch to adjacent STP switches and MST regions.

The following figure shows a network with three MST regions and an 802.1D switch (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.

**Figure 22: MST Regions, CIST Regional Roots, and CST Root**



Only the CST instance sends and receives BPDUs. MSTIs add their spanning tree information into the BPDUs (as M-records) to interact with neighboring switches and compute the final spanning tree topology. Because of this process, 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 switches use Version 3 BPDUs or 802.1D STP BPDUs to communicate with 802.1D-only switches. MST switches use MST BPDUs to communicate with MST switches.

## 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 switch to the CIST. The CIST external root path cost is the root path cost calculated between these virtual switches and switches 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 switch 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 switch 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 switch 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.

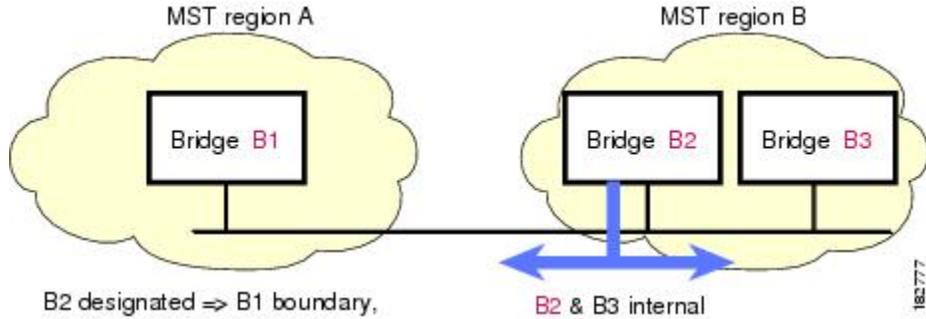
You configure a maximum aging time as the number of seconds that a switch waits without receiving spanning tree configuration messages before attempting a reconfiguration.

## Boundary Ports

A boundary port is a port that connects one region to another. 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 (see the following figure).

**Figure 23: 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.

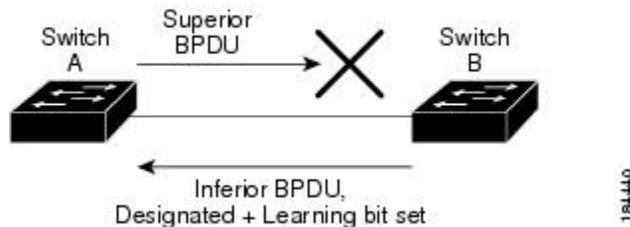
## Spanning-Tree Dispute Mechanism

Currently, this feature is not present in the IEEE MST standard, but it is included in the standard-compliant implementation. 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.

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.

The following 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. The block is shown as an STP dispute.

**Figure 24: Detecting a Unidirectional Link Failure**



## 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:

- 10 Mbps—2,000,000
- 100 Mbps—200,000
- 1 Gigabit Ethernet—20,000
- 10 Gigabit Ethernet—2,000

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 switch that runs MST supports a built-in protocol migration feature that enables it to interoperate with 802.1D STP switches. If this switch 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 switch 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 switch does not automatically revert to the MST mode if it no longer receives 802.1D BPDUs because it cannot detect whether the 802.1D switch has been removed from the link unless the 802.1D switch is the designated switch. A switch might also continue to assign a boundary role to a port when the switch to which this switch is connected has joined the region.

To restart the protocol migration process (force the renegotiation with neighboring switches), enter the **clear spanning-tree detected-protocols** command.

All Rapid PVST+ switches (and all 802.1D STP switches) on the link can process MST BPDUs as if they are 802.1w BPDUs. MST switches 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 switch of which is either a single spanning tree switch or a switch with a different MST configuration.

**Note**

---

MST interoperates with the Cisco prestandard Multiple Spanning Tree Protocol (MSTP) whenever it receives prestandard MSTP on an MST port; no explicit configuration is necessary.

---

## Interoperability with Rapid PVST+: Understanding PVST Simulation

MST interoperates with Rapid PVST+ with no need for user configuration. The PVST simulation feature enables this seamless interoperability.

**Note**

PVST simulation is enabled by default. That is, by default, all interfaces on the switch 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 Rapid PVST+-enabled port. Because Rapid PVST+ is the default STP mode, you may encounter many Rapid PVST+-enabled connections.

Disabling Rapid PVST+ simulation, which can be done per port or globally for the entire switch, 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.

## Configuring MST

### MST Configuration Guidelines

When configuring MST, follow these guidelines:

- When you work with private VLANs, enter the **private-vlan synchronize** command to map the secondary VLANs to the same MST instance as the primary VLAN.
- When you are in the MST configuration mode, 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 mode without committing any changes, enter the **abort** command.
  - To leave the MST configuration mode and commit all the changes that you made before you left the mode, enter the **exit** command.

### Enabling MST

You must enable MST; Rapid PVST+ is the default.

**Caution**

Changing the spanning tree mode disrupts traffic because all spanning tree instances are stopped for the previous mode and started for the new mode. Also, having two different spanning-tree modes on Virtual Port Channel (vPC) peer switches is an inconsistency, so this operation is disruptive.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch# **configure terminal**
3. switch(config)# **spanning-tree mode mst**
4. (Optional) switch(config)# **no spanning-tree mode mst**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch# <b>configure terminal</b>	Enters configuration mode.
<b>Step 3</b>	switch(config)# <b>spanning-tree mode mst</b>	Enables MST on the switch.
<b>Step 4</b>	switch(config)# <b>no spanning-tree mode mst</b>	(Optional) Disables MST on the switch and returns you to Rapid PVST+.

This example shows how to enable MST on the switch:

```
switch# configure terminal
switch(config)# spanning-tree mode mst
```



### Note

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

## Entering MST Configuration Mode

You enter MST configuration mode to configure the MST name, VLAN-to-instance mapping, and MST revision number on the switch.

For two or more switches to be 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.

When you are working in MST configuration mode, note the difference between the **exit** and **abort** commands.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mst configuration**
3. switch(config-mst)# **exit** or switch(config-mst)# **abort**
4. (Optional) switch(config)# **no spanning-tree mst configuration**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>spanning-tree mst configuration</b>	Enters MST configuration mode on the system. You must be in the MST configuration mode to assign the MST configuration parameters, as follows: <ul style="list-style-type: none"> <li>• MST name</li> <li>• Instance-to-VLAN mapping</li> <li>• MST revision number</li> <li>• Synchronize primary and secondary VLANs in private VLANs</li> </ul>
<b>Step 3</b>	switch(config-mst)# <b>exit</b> or switch(config-mst)# <b>abort</b>	Exits or aborts. <ul style="list-style-type: none"> <li>• The <b>exit</b> command commits all the changes and exits MST configuration mode.</li> <li>• The <b>abort</b> command exits the MST configuration mode without committing any of the changes.</li> </ul>
<b>Step 4</b>	switch(config)# <b>no spanning-tree mst configuration</b>	(Optional) Returns the MST region configuration to the following default values: <ul style="list-style-type: none"> <li>• The region name is an empty string.</li> <li>• No VLANs are mapped to any MST instance (all VLANs are mapped to the CIST instance).</li> <li>• The revision number is 0.</li> </ul>

## Specifying the MST Name

You configure a region name on the bridge. For two or more bridges to be in the same MST region, they must have the identical MST name, VLAN-to-instance mapping, and MST revision number.

**SUMMARY STEPS**

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mst configuration**
3. switch(config-mst)# **name name**

**DETAILED STEPS**

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>spanning-tree mst configuration</b>	Enters MST configuration submode.
<b>Step 3</b>	switch(config-mst)# <b>name name</b>	Specifies the name for MST region. The <i>name</i> string has a maximum length of 32 case-sensitive characters. The default is an empty string.

This example shows how to set the name of the MST region:

```
switch# configure terminal
switch(config)# spanning-tree mst configuration
switch(config-mst)# name accounting
```

**Specifying the MST Configuration Revision Number**

You configure the revision number on the bridge. For two or more bridges to be in the same MST region, they must have the identical MST name, VLAN-to-instance mapping, and MST revision number.

**SUMMARY STEPS**

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mst configuration**
3. switch(config-mst)# **revision version**

**DETAILED STEPS**

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>spanning-tree mst configuration</b>	Enters MST configuration submode.
<b>Step 3</b>	switch(config-mst)# <b>revision version</b>	Specifies the revision number for the MST region. The range is from 0 to 65535, and the default value is 0.

This example shows how to configure the revision number of the MSTI region for 5:

```
switch# configure terminal
switch(config)# spanning-tree mst configuration
switch(config-mst)# revision 5
```

## Specifying the Configuration on an MST Region

For two or more switches 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. switch# **configure terminal**
2. switch(config)# **spanning-tree mst configuration**
3. switch(config-mst)# **instance** *instance-id* **vlan** *vlan-range*
4. switch(config-mst)# **name** *name*
5. switch(config-mst)# **revision** *version*

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>spanning-tree mst configuration</b>	Enters MST configuration submode.
<b>Step 3</b>	switch(config-mst)# <b>instance</b> <i>instance-id</i> <b>vlan</b> <i>vlan-range</i>	<p>Maps VLANs to an MST instance as follows:</p> <ul style="list-style-type: none"> <li>• For <i>instance-id</i> , the range is from 1 to 4094.</li> <li>• For <b>vlan</b> <i>vlan-range</i> , the range is from 1 to 4094.</li> </ul> <p>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.</p> <p>To specify a VLAN range, enter a hyphen; for example, enter the <b>instance 1 vlan 1-63</b> command to map VLANs 1 through 63 to MST instance 1.</p> <p>To specify a VLAN series, enter a comma; for example, enter the <b>instance 1 vlan 10, 20, 30</b> command to map VLANs 10, 20, and 30 to MST instance 1.</p>

	Command or Action	Purpose
<b>Step 4</b>	switch(config-mst)# <b>name</b> <i>name</i>	Specifies the instance name. The <i>name</i> string has a maximum length of 32 case-sensitive characters.
<b>Step 5</b>	switch(config-mst)# <b>revision</b> <i>version</i>	Specifies the configuration revision number. The range is from 0 to 65535.

To return to defaults, do the following:

- To return to the default MST region configuration settings, enter the **no spanning-tree mst configuration** configuration command.
- To return to the default VLAN-to-instance map, enter the **no instance** *instance-id* **vlan** *vlan-range* MST configuration command.
- To return to the default name, enter the **no name** MST configuration command.
- To return to the default revision number, enter the **no revision** MST configuration command.
- To reenables Rapid PVST+, enter the **no spanning-tree mode** or the **spanning-tree mode rapid-pvst** global configuration command.

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)# 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)# show pending
Pending MST configuration
Name      [region1]
Revision  1
Instances configured 2
Instance  Vlans Mapped
-----
0         1-9,21-4094
1         10-20
-----
```

## Mapping and Unmapping VLANs to MST Instances



### Caution

When you change the VLAN-to-MSTI mapping, the system restarts MST.



### Note

You cannot disable an MSTI.

For two or more bridges to be in the same MST region, they must have the identical MST name, VLAN-to-instance mapping, and MST revision number.

**SUMMARY STEPS**

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mst configuration**
3. switch(config-mst)# **instance** *instance-id* **vlan** *vlan-range*
4. switch(config-mst)# **no instance** *instance-id* **vlan** *vlan-range*

**DETAILED STEPS**

	<b>Command or Action</b>	<b>Purpose</b>
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>spanning-tree mst configuration</b>	Enters MST configuration submode.
<b>Step 3</b>	switch(config-mst)# <b>instance</b> <i>instance-id</i> <b>vlan</b> <i>vlan-range</i>	Maps VLANs to an MST instance, as follows: <ul style="list-style-type: none"> <li>• For <i>instance-id</i> the range is from 1 to 4094. Instance 0 is reserved for the IST for each MST region.</li> <li>• For <i>vlan-range</i> the range is from 1 to 4094.</li> </ul> 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.
<b>Step 4</b>	switch(config-mst)# <b>no instance</b> <i>instance-id</i> <b>vlan</b> <i>vlan-range</i>	Deletes the specified instance and returns the VLANs to the default MSTI, which is the CIST.

This example shows how to map VLAN 200 to MSTI 3:

```
switch# configure terminal
switch(config)# spanning-tree mst configuration
switch(config-mst)# instance 3 vlan 200
```

**Mapping Secondary VLANs to Same MSTI as Primary VLANs for Private VLANs**

When you are working with private VLANs on the system, all secondary VLANs must be in the same MSTI and their associated primary VLAN.

**SUMMARY STEPS**

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mst configuration**
3. switch(config-mst)# **private-vlan synchronize**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	switch(config)# <b>spanning-tree mst configuration</b>	Enters MST configuration submenu.
Step 3	switch(config-mst)# <b>private-vlan synchronize</b>	Automatically maps all secondary VLANs to the same MSTI as their associated primary VLAN in all private VLANs.

This example shows how to automatically map all the secondary VLANs to the same MSTI as their associated primary VLANs in all private VLANs:

```
switch# configure terminal
switch(config)# spanning-tree mst configuration
switch(config-mst)# private-vlan synchronize
```

## Configuring the Root Bridge

You can configure the switch to become the root bridge.


**Note**

The root bridge for each MSTI should be a backbone or distribution switch. Do not configure an access switch as the spanning tree primary root bridge.

Enter the **diameter** keyword, which is available only for MSTI 0 (or the IST), to specify the network diameter (that is, the maximum number of hops between any two end stations in the network). When you specify the network diameter, the switch 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 switch 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. switch# **configure terminal**
2. switch(config)# **spanning-tree mst instance-id root {primary | secondary} [diameter dia [hello-time hello-time]]**
3. (Optional) switch(config)# **no spanning-tree mst instance-id root**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	switch(config)# <b>spanning-tree mst</b> <i>instance-id</i> <b>root</b> { <b>primary</b>   <b>secondary</b> } [ <b>diameter</b> <i>dia</i> [ <b>hello-time</b> <i>hello-time</i> ]]	Configures a switch as the root bridge as follows: <ul style="list-style-type: none"> <li>• For <i>instance-id</i>, 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.</li> <li>• For diameter <i>net-diameter</i>, specify the maximum number of hops between any two end stations. The default is 7. This keyword is available only for MST instance 0.</li> <li>• For hello-time <i>seconds</i>, 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.</li> </ul>
Step 3	switch(config)# <b>no spanning-tree mst</b> <i>instance-id</i> <b>root</b>	(Optional) Returns the switch priority, diameter, and hello time to default values.

This example shows how to configure the switch as the root switch for MSTI 5:

```
switch# configure terminal
switch(config)# spanning-tree mst 5 root primary
```

## Configuring a Secondary Root Bridge

You can execute this command on more than one switch 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** configuration command.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mst** *instance-id* **root** {**primary** | **secondary**} [**diameter** *dia* [**hello-time** *hello-time*]]
3. (Optional) switch(config)# **no spanning-tree mst** *instance-id* **root**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# <b>configure terminal</b>	Enters global configuration mode.

	Command or Action	Purpose
<b>Step 2</b>	switch(config)# <b>spanning-tree mst</b> <i>instance-id</i> <b>root</b> { <b>primary</b>   <b>secondary</b> } [ <b>diameter</b> <i>dia</i> [ <b>hello-time</b> <i>hello-time</i> ]]	Configures a switch as the secondary root bridge as follows: <ul style="list-style-type: none"> <li>• For <i>instance-id</i>, 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.</li> <li>• For diameter <i>net-diameter</i>, specify the maximum number of hops between any two end stations. The default is 7. This keyword is available only for MST instance 0.</li> <li>• For hello-time <i>seconds</i>, 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.</li> </ul>
<b>Step 3</b>	switch(config)# <b>no spanning-tree mst</b> <i>instance-id</i> <b>root</b>	(Optional) Returns the switch priority, diameter, and hello-time to default values.

This example shows how to configure the switch as the secondary root switch for MSTI 5:

```
switch# configure terminal
switch(config)# spanning-tree mst 5 root secondary
```

## Configuring the 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. switch# **configure terminal**
2. switch(config)# **interface** {{*type slot/port*} | {**port-channel number**}}
3. switch(config-if)# **spanning-tree mst** *instance-id* **port-priority** *priority*

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface</b> {{ <i>type slot/port</i> }   { <b>port-channel number</b> }}	Specifies an interface to configure, and enters interface configuration mode.
<b>Step 3</b>	switch(config-if)# <b>spanning-tree mst</b> <i>instance-id</i> <b>port-priority</b> <i>priority</i>	Configures the port priority as follows:

	Command or Action	Purpose
		<ul style="list-style-type: none"> <li>For <i>instance-id</i>, 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.</li> <li>For <i>priority</i>, the range is 0 to 224 in increments of 32. The default is 128. A lower number indicates a higher priority.</li> </ul> <p>The priority values are 0, 32, 64, 96, 128, 160, 192, and 224. The system rejects all other values.</p>

This example shows how to set the MST interface port priority for MSTI 3 on Ethernet port 3/1 to 64:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# spanning-tree mst 3 port-priority 64
```

You can only apply this command to a physical Ethernet interface.

## Configuring the Port Cost

The MST path-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. switch# **configure terminal**
2. switch(config)# **interface** *{{type slot/port} | {port-channel number}}*
3. switch(config-if)# **spanning-tree mst instance-id cost** *[cost | auto]*

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface</b> <i>{{type slot/port}   {port-channel number}}</i>	Specifies an interface to configure, and enters interface configuration mode.
<b>Step 3</b>	switch(config-if)# <b>spanning-tree mst instance-id cost</b> <i>[cost   auto]</i>	Configures the cost.

	Command or Action	Purpose
		<p>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:</p> <ul style="list-style-type: none"> <li>• For <i>instance-id</i>, 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.</li> <li>• For <i>cost</i>, the range is from 1 to 200000000. The default value is auto, which is derived from the media speed of the interface.</li> </ul>

This example shows how to set the MST interface port cost on Ethernet 3/1 for MSTI 4:

```
switch# configure terminal
switch(config)# interface ethernet 3/1
switch(config-if)# spanning-tree mst 4 cost 17031970
```

## Configuring the Switch Priority

You can configure the switch priority for an MST instance so that it is more likely that the specified switch is chosen as the root bridge.



### Note

Exercise care when using this 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. switch# **configure terminal**
2. switch(config)# **spanning-tree mst instance-id priority priority-value**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>spanning-tree mst instance-id priority priority-value</b>	<p>Configures a switch priority as follows:</p> <ul style="list-style-type: none"> <li>• For <i>instance-id</i>, 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.</li> </ul>

	Command or Action	Purpose
		<ul style="list-style-type: none"> <li>For priority, the range is from 0 to 61440 in increments of 4096; the default is 32768. A lower number indicates that the switch will most likely be chosen as the root bridge.</li> </ul> <p>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.</p>

This example shows how to configure the priority of the bridge to 4096 for MSTI 5:

```
switch# configure terminal
switch(config)# spanning-tree mst 5 priority 4096
```

## Configuring the Hello Time

You can configure the interval between the generation of configuration messages by the root bridge for all instances on the switch by changing the hello time.



### Note

Exercise care when using this 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** configuration commands to modify the hello time.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mst hello-time *seconds***

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>spanning-tree mst hello-time <i>seconds</i></b>	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 switch is alive. For <i>seconds</i> , the range is from 1 to 10, and the default is 2 seconds.

This example shows how to configure the hello time of the switch to 1 second:

```
switch# configure terminal
switch(config)# spanning-tree mst hello-time 1
```

## Configuring the Forwarding-Delay Time

You can set the forward delay timer for all MST instances on the switch with one command.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mst forward-time seconds**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>spanning-tree mst forward-time seconds</b>	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 <i>seconds</i> , the range is from 4 to 30, and the default is 15 seconds.

This example shows how to configure the forward-delay time of the switch to 10 seconds:

```
switch# configure terminal
switch(config)# spanning-tree mst forward-time 10
```

## Configuring the Maximum-Aging Time

The maximum-aging timer is the number of seconds that a switch waits without receiving spanning tree configuration messages before attempting a reconfiguration.

You set the maximum-aging timer for all MST instances on the switch with one command (the maximum age time only applies to the IST).

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mst max-age seconds**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	switch(config)# <b>spanning-tree mst max-age</b> <i>seconds</i>	Configures the maximum-aging time for all MST instances. The maximum-aging time is the number of seconds that a switch waits without receiving spanning tree configuration messages before attempting a reconfiguration. For <i>seconds</i> , the range is from 6 to 40, and the default is 20 seconds.

This example shows how to configure the maximum-aging timer of the switch to 40 seconds:

```
switch# configure terminal
switch(config)# spanning-tree mst max-age 40
```

## Configuring the Maximum-Hop Count

MST uses the path cost to the IST regional root and a hop-count mechanism similar to the IP time-to-live (TTL) mechanism. You 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).

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree mst max-hops** *hop-count*

## DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	switch(config)# <b>spanning-tree mst max-hops</b> <i>hop-count</i>	Specifies the number of hops in a region before the BPDU is discarded, and the information held for a port is aged. For <i>hop-count</i> , the range is from 1 to 255, and the default value is 20 hops.

This example shows how to set the maximum hops to 40:

```
switch# configure terminal
switch(config)# spanning-tree mst max-hops 40
```

## Configuring PVST Simulation Globally

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 switch while you are in interface command mode.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **no spanning-tree mst simulate pvst global**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>no spanning-tree mst simulate pvst global</b>	Disables all interfaces on the switch from automatically interoperating with connected switch that is running in Rapid PVST+ mode. By default, all interfaces on the switch operate seamlessly between Rapid PVST+ and MST.

This example shows how to prevent the switch from automatically interoperating with a connecting switch that is running Rapid PVST+:

```
switch# configure terminal
switch(config)# no spanning-tree mst simulate pvst global
```

## Configuring PVST Simulation Per Port

MST interoperates seamlessly with Rapid PVST+. However, to prevent an accidental connection to a switch 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.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *{{type slot/port} | {port-channel number}}*
3. switch(config-if)# **spanning-tree mst simulate pvst disable**
4. switch(config-if)# **spanning-tree mst simulate pvst**
5. switch(config-if)# **no spanning-tree mst simulate pvst**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	switch(config)# <b>interface</b> <i>{{type slot/port}   {port-channel number}}</i>	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# <b>spanning-tree mst simulate pvst disable</b>	Disables specified interfaces from automatically interoperating with a connected switch that is running in Rapid PVST+ mode.  By default, all interfaces on the switch operate seamlessly between Rapid PVST+ and MST.
Step 4	switch(config-if)# <b>spanning-tree mst simulate pvst</b>	Reenables the seamless operation between MST and Rapid PVST+ on specified interfaces.
Step 5	switch(config-if)# <b>no spanning-tree mst simulate pvst</b>	Sets the interface to the switch-wide MST and Rapid PVST+ interoperation that you configured using the <b>spanning-tree mst simulate pvst global</b> command.

This example shows how to prevent the specified interfaces from automatically interoperating with a connecting switch that is not running MST:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree mst simulate pvst disable
```

## Specifying the Link Type

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 switch, you can override the default setting on the link type and enable rapid transitions.

If you set the link to shared, STP reverts to 802.1D.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot/port*
3. switch(config-if)# **spanning-tree link-type** {*auto | point-to-point | shared*}

## DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# <b>configure terminal</b>	Enters configuration mode.
Step 2	switch(config)# <b>interface</b> <i>type slot/port</i>	Specifies the interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# <b>spanning-tree link-type</b> { <b>auto</b>   <b>point-to-point</b>   <b>shared</b> }	Configures the link type to be either point to point or shared. The system reads the default value from the switch connection. Half-duplex links are shared and full-duplex links are point to point. If the link type is shared, the STP reverts to 802.1D. The default is auto, which sets the link type based on the duplex setting of the interface.

This example shows how to configure the link type as point to point:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree link-type point-to-point
```

## Restarting the Protocol

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 switch, which is a switch that runs only IEEE 802.1D, has been removed from the link unless the legacy switch is the designated switch. Enter this command to restart the protocol negotiation (force the renegotiation with neighboring switches) on the entire switch or on specified interfaces.

## SUMMARY STEPS

1. switch# **clear spanning-tree detected-protocol** [**interface** *interface* [*interface-num* | *port-channel*]]

## DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# <b>clear spanning-tree detected-protocol</b> [ <b>interface</b> <i>interface</i> [ <i>interface-num</i>   <i>port-channel</i> ]]	Restarts MST on the entire switch or specified interfaces.

This example shows how to restart MST on the Ethernet interface on slot 2, port 8:

```
switch# clear spanning-tree detected-protocol interface ethernet 2/8
```

## Verifying the MST Configuration

Use the following commands to display MST configuration information.

Command	Purpose
<code>show running-config spanning-tree [all]</code>	Displays the current spanning tree configuration.
<code>show spanning-tree mst [options]</code>	Displays detailed information for the current MST configuration.

This example shows how to display the current MST configuration:

```
switch# show spanning-tree mst configuration
% Switch is not in mst mode
Name      [mist-attempt]
Revision  1      Instances configured 2
Instance  Vlans mapped
-----
0         1-12,14-41,43-4094
1         13,42
```





## Configuring STP Extensions

This chapter contains the following sections:

- [Overview, page 215](#)

### Overview

Cisco has added extensions to Spanning Tree Protocol (STP) that make convergence more efficient. In some cases, even though 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 can be used with both RPVST+ and Multiple Spanning Tree Protocol (MST).

The available extensions are spanning tree port types, Bridge Assurance, BPDU Guard, BPDU Filtering, Loop Guard, and Root Guard. 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.

### Information About STP Extensions

#### Understanding 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. Depending on the type of device to which the interface is connected, you can configure a spanning tree port as one of these port types.

#### Spanning Tree Edge Ports

Edge ports, which are connected to hosts, can be either an access port or a trunk port. 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 hosts should not receive STP bridge protocol data units (BPDUs).



---

**Note** If you configure a port connected to another switch as an edge port, you might create a bridging loop.

---

### Spanning Tree Network Ports

Network ports are connected only to switches or bridges. Configuring a port as a network port while Bridge Assurance is enabled globally, enables Bridge Assurance on that port.



---

**Note** If you mistakenly configure ports that are connected to hosts or other edge devices as spanning tree network ports, those ports automatically move into the blocking state.

---

### Spanning Tree Normal Ports

Normal ports can be connected to either hosts, switches, or bridges. These ports function as normal spanning tree ports.

The default spanning tree interface is a normal port.

## Understanding 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 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. Legacy 802.1D spanning tree does not support Bridge Assurance.

---

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.

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.

## Understanding 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, LAN edge interfaces do not receive BPDUs. A BPDU that is received by an edge LAN

interface signals an invalid configuration, such as the connection of an unauthorized host or switch. 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 LAN interface back in service after an invalid configuration.



**Note** When enabled globally, BPDU Guard applies to all operational spanning tree edge interfaces.

## Understanding BPDU Filtering

You can use BPDU Filtering to prevent the switch 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 configuring BPDU Filtering on a port that is not connected to a host, it can result in bridging loops because the port ignores any BPDU that it receives and goes to forwarding.

If the port configuration is not set to default BPDU Filtering, the edge configuration does not affect BPDU Filtering. The following table lists all the BPDU Filtering combinations.

**Table 14: BPDU Filtering Configurations**

BPDU Filtering Per Port Configuration	BPDU Filtering Global Configuration	STP Edge Port Configuration	BPDU Filtering State
Default	Enabled	Enabled	EnabledThe 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.
Default	Enabled	Disabled	Disabled
Default	Disabled	Enabled/Disabled	Disabled

BPDU Filtering Per Port Configuration	BPDU Filtering Global Configuration	STP Edge Port Configuration	BPDU Filtering State
Disable	Enabled/Disabled	Enabled/Disabled	Disabled
Enabled	Enabled/Disabled	Enabled/Disabled	Enabled <b>Caution</b> BPDUs are never sent and if received, they do not trigger the regular STP behavior - use with caution.

## Understanding Loop Guard

Loop Guard protects networks from loops that are caused by the following:

- Network interfaces that malfunction
- Busy CPUs
- Anything that prevents the normal forwarding of BPDUs

An STP loop occurs when a blocking port in a redundant topology erroneously transitions to the forwarding state. This transition usually happens because one of the ports in a physically redundant topology (not necessarily the blocking port) stops receiving BPDUs.

Loop Guard 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.



### Note

Loop Guard can be enabled only on network and normal spanning tree port types.

You can use Loop Guard to determine if a root port or an alternate/backup root port receives BPDUs. If the port does not receive BPDUs, Loop Guard puts the port into an inconsistent state (blocking) until the port starts to receive BPDUs again. A port in the inconsistent state does not transmit BPDUs. If the port receives BPDUs again, the protocol removes its loop-inconsistent condition, and the STP determines the port state because such 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.

## Understanding 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 sending superior BPDUs, the port is unblocked again. Through STP, the port moves to the forwarding state. Recovery is automatic.

Root Guard enabled on an interface applies this functionality 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.

**Note**

---

You can enable Root Guard on all spanning tree port types: normal, edge, and network ports.

---

## Configuring STP Extensions

### STP Extensions Configuration Guidelines

When configuring STP extensions, follow these guidelines:

- Configure all access and trunk ports connected to 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.
- Loop Guard does not run on spanning tree edge ports.
- Enabling Loop Guard on ports that are not connected to a point-to-point link will not work.
- You cannot enable Loop Guard if Root Guard is enabled.

### Configuring Spanning Tree Port Types Globally

The spanning tree port type designation depends on the type of device the port is connected to, as follows:

- Edge—Edge ports are connected to hosts and can be either an access port or a trunk port.
- Network—Network ports are connected only to switches or bridges.
- Normal—Normal ports are neither edge ports nor network ports; they are normal spanning tree ports. These ports can be connected to any type of device.

You can configure the port type either globally or per interface. By default, the spanning tree port type is normal.

**Before You Begin**

Ensure that STP is configured.

Ensure that you are configuring the ports correctly for the type of device to which the interface is connected.

**SUMMARY STEPS**

1. switch# **configure terminal**
2. switch(config)# **spanning-tree port type edge default**
3. switch(config)# **spanning-tree port type network default**

**DETAILED STEPS**

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>spanning-tree port type edge default</b>	Configures all interfaces as edge ports. Using this command assumes all ports are connected to hosts/servers. 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.
<b>Step 3</b>	switch(config)# <b>spanning-tree port type network default</b>	Configures all interfaces as spanning tree network ports. Using this command assumes all ports are connected to switches and bridges. If you enable Bridge Assurance, it automatically runs on network ports. By default, spanning tree ports are normal port types.  <b>Note</b> If you configure interfaces connected to hosts as network ports, those ports automatically move into the blocking state.

This example shows how to configure all access and trunk ports connected to hosts as spanning tree edge ports:

```
switch# configure terminal
switch(config)# spanning-tree port type edge default
```

This example shows how to configure all ports connected to switches or bridges as spanning tree network ports:

```
switch# configure terminal
switch(config)# spanning-tree port type network default
```

**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 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 disable** command.

**Before You Begin**

Ensure that STP is configured.  
 Ensure that the interface is connected to hosts.

**SUMMARY STEPS**

1. switch# **configure terminal**
2. switch(config)# **interface type slot/port**
3. switch(config-if)# **spanning-tree port type edge**

**DETAILED STEPS**

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface type slot/port</b>	Specifies the interface to configure, and enters the interface configuration mode.
<b>Step 3</b>	switch(config-if)# <b>spanning-tree port type edge</b>	Configures the specified access interfaces to be spanning 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.

This example shows how to configure the Ethernet access interface 1/4 to be a spanning tree edge port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree port type edge
```

## 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 a host that is configured as a network port automatically moves into the blocking state.

### Before You Begin

Ensure that STP is configured.

Ensure that the interface is connected to switches or routers.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** type slot/port
3. switch(config-if)# **spanning-tree port type network**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface</b> type slot/port	Specifies the interface to configure, and enters the interface configuration mode. The interface can be a physical Ethernet port.
<b>Step 3</b>	switch(config-if)# <b>spanning-tree port type network</b>	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.

This example shows how to configure the Ethernet interface 1/4 to be a spanning tree network port:

```
switch# configure terminal
switch(config)# interface ethernet 1/4
switch(config-if)# spanning-tree port type network
```

## 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

Ensure that STP is configured.

Ensure that you have configured some spanning tree edge ports.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree port type edge bpduguard default**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	switch(config)# <b>spanning-tree port type edge bpduguard default</b>	Enables BPDU Guard by default on all spanning tree edge ports. By default, global BPDU Guard is disabled.

This example shows how to enable BPDU Guard on all spanning tree edge ports:

```
switch# configure terminal
switch(config)# spanning-tree port type edge bpduguard default
```

## 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

Ensure that STP is configured.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot/port*
3. switch(config-if)# **spanning-tree bpduguard** {enable | disable}
4. (Optional) switch(config-if)# **no spanning-tree bpduguard**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface</b> <i>type slot/port</i>	Specifies the interface to configure, and enters the interface configuration mode.
<b>Step 3</b>	switch(config-if)# <b>spanning-tree bpduguard</b> {enable   disable}	Enables or disables BPDU Guard for the specified spanning tree edge interface. By default, BPDU Guard is disabled on physical Ethernet interfaces.
<b>Step 4</b>	switch(config-if)# <b>no spanning-tree bpduguard</b>	(Optional) Disables BPDU Guard on the interface.  <b>Note</b> Enables BPDU Guard on the interface if it is an operational edge port and if you enter the <b>spanning-tree port type edge bpduguard default</b> command.

This example shows how to explicitly enable BPDU Guard on the Ethernet edge port 1/4:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree bpduguard enable
switch(config-if)# no spanning-tree bpduguard
```

## 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 and 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 it incorrectly can cause bridging loops.



**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.

**Before You Begin**

Ensure that STP is configured.

Ensure that you have configured some spanning tree edge ports.

**SUMMARY STEPS**

1. switch# **configure terminal**
2. switch(config)# **spanning-tree port type edge bpdudfilter default**

**DETAILED STEPS**

	<b>Command or Action</b>	<b>Purpose</b>
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>spanning-tree port type edge bpdudfilter default</b>	Enables BPDU Filtering by default on all operational spanning tree edge ports. Global BPDU Filtering is disabled by default.

This example shows how to enable BPDU Filtering on all operational spanning tree edge ports:

```
switch# configure terminal
switch(config)# spanning-tree port type edge bpdudfilter default
```

**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 bpdudfilter 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 ignores any BPDU it receives and goes to forwarding.

You can enter this command to override the port configuration on specified interfaces.

This command has three states:

- **spanning-tree bpdudfilter enable**—Unconditionally enables BPDU Filtering on the interface.
- **spanning-tree bpdudfilter disable**—Unconditionally disables BPDU Filtering on the interface.
- **no spanning-tree bpdudfilter**—Enables BPDU Filtering on the interface if the interface is an operational edge port and if you configure the **spanning-tree port type edge bpdudfilter default** command.



**Note** When you enable BPDU Filtering locally on a port, this feature prevents the device from receiving or sending BPDUs on this port.

### Before You Begin

Ensure that STP is configured.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot/port*
3. switch(config-if)# **spanning-tree bpdudfilter** {enable | disable}
4. (Optional) switch(config-if)# **no spanning-tree bpdudfilter**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface</b> <i>type slot/port</i>	Specifies the interface to configure, and enters the interface configuration mode.
<b>Step 3</b>	switch(config-if)# <b>spanning-tree bpdudfilter</b> {enable   disable}	Enables or disables BPDU Filtering for the specified spanning tree edge interface. By default, BPDU Filtering is disabled.
<b>Step 4</b>	switch(config-if)# <b>no spanning-tree bpdudfilter</b>	(Optional) Disables BPDU Filtering on the interface. <b>Note</b> Enables BPDU Filtering on the interface if the interface is an operational spanning tree edge port and if you enter the <b>spanning-tree port type edge bpdudfilter default</b> command.

This example shows how to explicitly enable BPDU Filtering on the Ethernet spanning tree edge port 1/4:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch (config-if)# spanning-tree bpdudfilter enable
```

## 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

Ensure that STP is configured.

Ensure that you have spanning tree normal ports or have configured some network ports.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **spanning-tree loopguard default**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	switch(config)# <b>spanning-tree loopguard default</b>	Enables Loop Guard by default on all spanning tree normal and network ports. By default, global Loop Guard is disabled.

This example shows how to enable Loop Guard on all spanning tree normal or network ports:

```
switch# configure terminal
switch(config)# spanning-tree loopguard default
```

## Enabling Loop Guard or Root Guard on Specified Interfaces

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 LoopGuard 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**

Ensure that STP is configured.

Ensure that you are configuring Loop Guard on spanning tree normal or network ports.

**SUMMARY STEPS**

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot/port*
3. switch(config-if)# **spanning-tree guard** {loop | root | none}

**DETAILED STEPS**

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface</b> <i>type slot/port</i>	Specifies the interface to configure, and enters the interface configuration mode.
<b>Step 3</b>	switch(config-if)# <b>spanning-tree guard</b> {loop   root   none}	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.  <b>Note</b> Loop Guard runs only on spanning tree normal and network interfaces.

This example shows how to enable Root Guard on Ethernet port 1/4:

```
switch# configure terminal
switch (config)# interface ethernet 1/4
switch(config-if)# spanning-tree guard root
```

**Verifying the STP Extension Configuration**

Use the following commands to display the configuration information for the STP extensions.

Command	Purpose
<b>show running-config spanning-tree</b> [all]	Displays the current status of spanning tree on the switch.
<b>show spanning-tree</b> [options]	Displays selected detailed information for the current spanning tree configuration.



## Configuring Flex Links

This chapter contains the following sections:

- [Information About Flex Links, page 229](#)
- [Guidelines and Limitations for Flex Link, page 230](#)
- [Default Settings for Flex Link, page 231](#)
- [Configuring Flex Links, page 232](#)
- [Configuring Flex Link Preemption, page 233](#)
- [Verifying Flex Link Configuration, page 235](#)
- [Flex Link Configuration Examples, page 236](#)

### Information About Flex Links

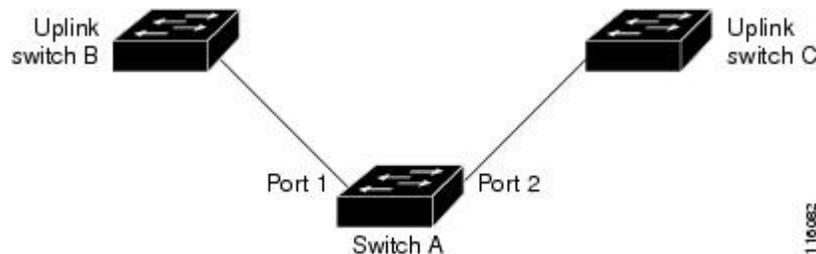
Flex links are a pair of a Layer 2 interfaces (switch ports or port channels) where one interface is configured to act as a backup to the other. The feature provides an alternative solution to the Spanning Tree Protocol (STP). You can disable STP and still retain basic link redundancy. Flex links are typically configured in service provider or enterprise networks where customers do not want to run STP on the switch. If the switch is running STP, Flex Links are not necessary because STP already provides link-level redundancy or backup.

You configure Flex Links on one Layer 2 interface (the active link) by assigning another Layer 2 interface as the Flex Links or backup link. The Flex Links interface can be on the same switch. 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. Flex Links are not configured by default and there are no backup interfaces defined. STP is disabled on Flex Link interfaces.

In the 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.

Flex Links are supported only on Layer 2 ports and port channels, not on VLANs or on Layer 3 ports.

**Figure 25: Flex Links Configuration Example**



### Preemption

You can optionally configure a preemption mechanism to specify the preferred port for forwarding traffic. For example, you can configure a Flex Link pair with preemption mode so that when a port comes back up, if it has greater bandwidth than the peer port, then it will begin forwarding after 35 seconds (default preemption delay) and the peer port will be on standby. This is done by entering the preemption mode bandwidth and delay commands.

If a primary (forwarding) link goes down, a trap notifies the network management stations. If the standby link goes down, a trap notifies the users.

You can configure preemption in the following three modes:

- Forced—The active interface always preempts the backup.
- Bandwidth—The interface with the higher bandwidth always acts as the active interface.
- Off—There is no preemption; the first interface that is up is put in forwarding mode.

You can also configure the preemption delay as a specified amount of time (in seconds) before preempting a working interface for another. This ensures that the counterpart in the upstream switch has transitioned to an STP forwarding state before the switch over.

### Multicast Fast-Convergence

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 for Flex Link

Consider the following guidelines and limitations when configuring Flex Links:

- 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.
- 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.
- Do not configure any STP features (for example, PortFast, BPDU Guard, and so forth) on Flex Links ports or the ports to which the links connect.
- 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.
- MVR configuration on Flex Link ports is not supported.

Flex links cannot be configured on the following interface types:

- FEX fabric ports and FEX host ports
- FCoE (vFC) interfaces
- VNTAG (vETH) interfaces
- Interfaces with port security enabled
- Layer 3 interfaces
- SPAN destinations
- Port channel members
- Interfaces configured with Private VLANs
- Interfaces in end node mode
- Fabric path core interfaces (Layer 2 multipath)

## Default Settings for Flex Link

*Table 15: Flex Link Default Parameter Settings*

Parameter	Definition
Multicast Fast-Convergence	Disabled
Preemption mode	Off
Preemption delay	35 seconds

# Configuring Flex Links

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.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config) # **feature flexlink**
3. switch(config) # **interface** { **ethernet** *slot/port* | **port-channel** *channel-no* }
4. switch(config-if) # **switchport backup interface** { **ethernet** *slot/port* | **port-channel** *channel-no* } [ **multicast fast-convergence** | **preemption** { **delay** *delay-time* | **mode** [ **bandwidth** | **forced** | **off** ] } ]
5. (Optional) switch(config-if) # **end**
6. (Optional) switch# **show interface** *interface-id* **switchport backup**
7. (Optional) switch# **copy running-config startup-config**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config) # <b>feature flexlink</b>	Enables Flex Link.
<b>Step 3</b>	switch(config) # <b>interface</b> { <b>ethernet</b> <i>slot/port</i>   <b>port-channel</b> <i>channel-no</i> }	Specifies the Ethernet or port channel interface and enters interface configuration mode.  The port channel range is 1 to 48.
<b>Step 4</b>	switch(config-if) # <b>switchport backup interface</b> { <b>ethernet</b> <i>slot/port</i>   <b>port-channel</b> <i>channel-no</i> } [ <b>multicast fast-convergence</b>   <b>preemption</b> { <b>delay</b> <i>delay-time</i>   <b>mode</b> [ <b>bandwidth</b>   <b>forced</b>   <b>off</b> ] } ]	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. <ul style="list-style-type: none"> <li>• <b>ethernet</b> <i>slot/port</i>—Specifies the backup Ethernet interface. The <i>slot</i> number is from 1 to 255 and the <i>port</i> number is from 1 to 128.</li> <li>• <b>port-channel</b> <i>port-channel-no</i>—Specifies the backup port channel interface. The <i>port-channel-no</i> number is from 1 to 4096.</li> <li>• <b>multicast</b>—Specifies the multicast parameters.</li> <li>• <b>fast-convergence</b>—Configures fast convergence on the backup interface.</li> <li>• <b>preemption</b>—Configures a preemption scheme for a backup interface pair.</li> <li>• <b>delay</b> <i>delay-time</i>—Specifies the preemption delay. The <i>delay-time</i> range is from 1 to 300 seconds. The default is 35 seconds.</li> <li>• <b>mode</b>—Specifies the preemption mode.</li> </ul>

	Command or Action	Purpose
		<ul style="list-style-type: none"> <li>• <b>bandwidth</b>—Specifies that the interface with the higher available bandwidth always preempts the backup.</li> <li>• <b>forced</b>—Specifies the interface that always preempts the backup.</li> <li>• <b>off</b>—Specifies that no preemption occurs from backup to active.</li> </ul>
<b>Step 5</b>	switch(config-if) # end	(Optional) Return to privileged EXEC mode.
<b>Step 6</b>	switch# show interface <i>interface-id</i> switchport backup	(Optional) Verifies the configuration.
<b>Step 7</b>	switch# copy running-config startup-config	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

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 ethernet1/1
switch(config-if)# switchport backup interface ethernet2/1
switch(config-if) # exit
switch(config)# interface po300
Switch(config-if)# switchport backup interface po301
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

You can configure a preemption scheme for a pair of Flex Links.

### Before You Begin

Enable the Flex Link feature.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config) # **interface ethernet 1/48 slot/port**
3. switch(config-if) # **switchport backup interface ethernet slot/port**
4. switch(config-if) # **switchport backup interface ethernet slot/port preempt mode [ bandwidth | forced | off ]**
5. switch(config-if) # **switchport backup interface ethernet slot/port preempt delay delay-time**
6. (Optional) switch(config-if) # **end**
7. (Optional) switch# **show interface interface-id switchport backup**
8. (Optional) switch# **copy running-config startup-config**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config) # <b>interface ethernet 1/48 slot/port</b>	Specifies the Ethernet interface and enters interface configuration mode.  The <b>interface</b> is a physical Layer 2 interface or a port channel (logical interface).  The <i>slot/port</i> range is from 1 to 48.
<b>Step 3</b>	switch(config-if) # <b>switchport backup interface ethernet slot/port</b>	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.
<b>Step 4</b>	switch(config-if) # <b>switchport backup interface ethernet slot/port preempt mode [ bandwidth   forced   off ]</b>	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.  Configure a preemption mechanism and delay for a Flex link interface pair. You can configure the preemption as: <ul style="list-style-type: none"> <li>• <b>bandwidth</b>—Interface with higher bandwidth always acts as the active interface</li> <li>• <b>forced</b>—Active interface always preempts the backup</li> <li>• <b>off</b>—No preemption happens from active to backup</li> </ul>
<b>Step 5</b>	switch(config-if) # <b>switchport backup interface ethernet slot/port preempt delay delay-time</b>	Configure the delay time until a port preempts another port. The default preemption delay is 35 seconds. <b>Note</b> Setting a delay time only works with forced and bandwidth modes.
<b>Step 6</b>	switch(config-if) # <b>end</b>	(Optional) Return to privileged EXEC mode.

	Command or Action	Purpose
Step 7	switch# <b>show interface <i>interface-id</i> switchport backup</b>	(Optional) Verifies the configuration.
Step 8	switch# <b>copy running-config startup-config</b>	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

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

```
Switch# configure terminal
switch(conf)# interface ethernet0/1
switch(conf-if)#switchport backup interface ethernet0/2 preemption mode forced
switch(conf-if)#switchport backup interface ethernet0/2 preemption delay 50
switch(conf-if)# end

switch# show interface switchport backup detail
Active Interface Backup Interface State
-----
Ethernet0/21 Ethernet0/2 Active Up/Backup Standby
Interface Pair : Gi0/1, Gi0/2
Preemption Mode : forced
Preemption Delay : 50 seconds
Bandwidth : 100000 Kbit (Gi0/1), 100000 Kbit (Gi0/2)
Mac Address Move Update Vlan : auto
```

## Verifying Flex Link Configuration

Use the following commands to display flex link configuration information:

Command	Purpose
<b>show interface switchport backup</b>	Displays information about all switch ort Flex Link interfaces.
<b>show interface switchport backup detail</b>	Displays detailed information about all switch ort Flex Link interfaces.
<b>show running-config backup</b> <b>show startup-config backup</b>	Displays the running or startup configuration for backup interfaces.
<b>show running-config flexlink</b> <b>show startup-config flexlink</b>	Displays the running or startup configuration for flex link interfaces.

## Flex Link Configuration Examples

This example shows how to configure a port-channel switchport backup pair with forced preemption. The active interface port-channel10 is the preferred forwarding interface:

```
switch(config)# interface port-channel10
switch(config-if)# switchport backup interface port-channel20 preemption mode forced
switch(config-if)# switchport backup interface port-channel20 preemption delay 35
```

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: po300 and po301. A general query received on po300 makes it an mrouter port and po301 as co-learned.

```
switch(config)# interface po300
Switch(config-if)# switchport backup interface po301
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
```

This example shows po300 and po301 as mrouter ports (po301 is co-learned); it is not added to the hardware table when multicast fast-convergence is disabled.

```
switch# show ip igmp snooping groups vlan 4
Type: S - Static, D - Dynamic, R - Router port
```

```
Vlan Group Address Ver Type Port list
4 */* - R Po300 Po301
224.1.1.1 v2 D Eth1/31
```

```
switch# show platform fwm info hw-stm | grep 0100.5e01.0101
1.4 0100.5e01.0101 midx 36 1:2849:0 0:0:1:0 1.0.0.0.0.24 (e:0)
```

```
switch# show platform fwm info oifl 36
oifl 36 vdc 1 oifl 36: vdc 1 gpinif 0, mcast idx 36(alt:0), oifl_type 2
oifl 36 vdc 1 oifl 36: oifl iods 8,44
oifl 36 vdc 1 oifl 36: max_iod 8192, ref count 1000 num_oifs 2, seq_num 33
oifl 36 vdc 1 oifl 36: hw pgmd: 1 msg present: 0
oifl 36 vdc 1 oifl 36: l2_bum_ref_cnt 0, l3_macg_ref_cnt 1000
oifl 36 vdc 1 oifl 36: if_indexs - Po300 Eth1/31
```

This example shows co-learned po301 is added to hardware when multicast fast-convergence is enabled:

```
switch(config)# interface po300
Switch(config-if)# switchport backup interface po301 multicast fast-convergence
```

```
switch# show platform fwm info hw-stm | grep 0100.5e01.0101
1.4 0100.5e01.0101 midx 38 1:2849:0 0:0:1:0 1.0.0.0.0.26 (e:0)
```

```
switch# show platform fwm info oifl 38
oifl 38 vdc 1 oifl 38: vdc 1 gpinif 0, mcast idx 38(alt:0), oifl_type 2
oifl 38 vdc 1 oifl 38: oifl iods 8-9,44
oifl 38 vdc 1 oifl 38: max_iod 8192, ref count 1000 num_oifs 3, seq_num 35
oifl 38 vdc 1 oifl 38: hw pgmd: 1 msg present: 0
oifl 38 vdc 1 oifl 38: l2_bum_ref_cnt 0, l3_macg_ref_cnt 1000
oifl 38 vdc 1 oifl 38: if_indexs - Po300 Po301 Eth1/31
```

This example shows the running configuration of Flex Link:

```
switch# show running-config flexlink

!Command: show running-config flexlink
!Time: Thu Jan 1 03:21:12 2011

version 5.0(3)N2(1)
feature flexlink
```

```
logging level Flexlink 5

interface port-channel500
  switchport backup interface port-channel501 preemption delay 36
  switchport backup interface port-channel501 multicast fast-convergence

interface Ethernet2/2
  switchport backup interface port-channel507 preemption mode forced
```

This example shows details about the Flex Link interface. Forced preemption is about to take place because (scheduled) is displayed.

```
switch# show interface switchport backup detail
```

```
Switch Backup Interface Pairs:
```

Active Interface	Backup Interface	State
-----		
port-channel300	port-channel301	Active Down/Backup Up
	Preemption Mode	: forced
	Preemption Delay	: 35 seconds (default) <b>(scheduled)</b>
	Multicast Fast Convergence	: Off
	Bandwidth	: 20000000 Kbit (port-channel300), 10000000 Kbit (port-channel301)





## Configuring LLDP

This chapter contains the following sections:

- [Configuring LLDP, page 239](#)
- [Configuring Interface LLDP, page 240](#)

## Configuring LLDP

### Before You Begin

Ensure that the Link Layer Discovery Protocol (LLDP) feature is enabled on the switch.

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **lldp** {**holdtime** *seconds* | **reinit** *seconds* | **timer** *seconds* | **tlv-select** {**dcb** | **management-address** | **port-description** | **port-vlan** | **system-capabilities** | **system-description** | **system-name**}}
3. switch(config)# **no lldp** {**holdtime** | **reinit** | **timer**}
4. (Optional)switch# **show lldp**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>lldp</b> { <b>holdtime</b> <i>seconds</i>   <b>reinit</b> <i>seconds</i>   <b>timer</b> <i>seconds</i>   <b>tlv-select</b> { <b>dcb</b>   <b>management-address</b>   <b>port-description</b>   <b>port-vlan</b>   <b>system-capabilities</b>   <b>system-description</b>   <b>system-name</b> }}	Configures LLDP options.  Use the <b>holdtime</b> option to set the length of time (10 to 255 seconds) that a device should save LLDP information received before discarding it. The default value is 120 seconds.  Use the <b>reinit</b> option to set the length of time (1 to 10 seconds) to wait before performing LLDP initialization on any interface. The default value is 2 seconds.

	Command or Action	Purpose
		<p>Use the <b>timer</b> option to set the rate (5 to 254 seconds) at which LLDP packets are sent. The default value is 30 seconds.</p> <p>Use the <b>tlv-select</b> option to specify the type length value (TLV). The default is enabled to send and receive all TLVs.</p> <p>Use the <b>dcbpx</b> option to specify the Data Center Ethernet Parameter Exchange (DCBXP) TLV messages.</p> <p>Use the <b>management-address</b> option to specify the management address TLV messages.</p> <p>Use the <b>port-description</b> option to specify the port description TLV messages.</p> <p>Use the <b>port-vlan</b> option to specify the port VLAN ID TLV messages.</p> <p>Use the <b>system-capabilities</b> option to specify the system capabilities TLV messages.</p> <p>Use the <b>system-description</b> option to specify the system description TLV messages.</p> <p>Use the <b>system-name</b> option to specify the system name TLV messages.</p>
<b>Step 3</b>	switch(config)# <b>no lldp</b> {holdtime   reinit   timer}	Resets the LLDP values to their defaults.
<b>Step 4</b>	(Optional)switch# <b>show lldp</b>	Displays LLDP configurations.

This example shows how to configure the global LLDP hold time to 200 seconds:

```
switch# configure terminal
switch(config)# lldp holdtime 200
switch(config)#
```

This example shows how to enable LLDP to send or receive the management address TLVs:

```
switch# configure terminal
switch(config)# lldp tlv-select management-address
switch(config)#
```

## Configuring Interface LLDP

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot/port*
3. switch(config-if)# [**no**] **lldp** {receive | transmit}
4. (Optional) switch# **show lldp** {interface | neighbors [**detail** | interface | system-detail] | timers | traffic}

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface type slot/port</b>	Selects the interface to change.
<b>Step 3</b>	switch(config-if)# [ <b>no</b> ] <b>lldp {receive   transmit}</b>	Sets the selected interface to either receive or transmit.  The <b>no</b> form of the command disables the LLDP transmit or receive.
<b>Step 4</b>	switch# <b>show lldp {interface   neighbors [detail   interface   system-detail]   timers   traffic}</b>	(Optional) Displays LLDP configurations.

This example shows how to set an interface to transmit LLDP packets:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# lldp transmit
```

This example shows how to configure an interface to disable LLDP:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# no lldp transmit
switch(config-if)# no lldp receive
```

This example shows how to display LLDP interface information:

```
switch# show lldp interface ethernet 1/2
tx_enabled: TRUE
rx_enabled: TRUE
dcbx_enabled: TRUE
Port MAC address: 00:0d:ec:a3:5f:48
Remote Peers Information
No remote peers exist
```

This example shows how to display LLDP neighbor information:

```
switch# show lldp neighbors
LLDP Neighbors
Remote Peers Information on interface Eth1/40
Remote peer's MSAP: length 12 Bytes:
00 c0 dd 0e 5f 3a 00 c0 dd 0e 5f 3a
LLDP TLV's
LLDP TLV type:Chassis ID LLDP TLV Length: 7
LLDP TLV type:Port ID LLDP TLV Length: 7
LLDP TLV type:Time to Live LLDP TLV Length: 2
LLDP TLV type:LLDP Organizationally Specific LLDP TLV Length: 55
LLDP TLV type:LLDP Organizationally Specific LLDP TLV Length: 5
LLDP TLV type:END of LLDPDU LLDP TLV Length: 0
Remote Peers Information on interface Eth1/34
Remote peer's MSAP: length 12 Bytes:
00 0d ec a3 27 40 00 0d ec a3 27 69
LLDP TLV's
LLDP TLV type:Chassis ID LLDP TLV Length: 7
LLDP TLV type:Port ID LLDP TLV Length: 7
LLDP TLV type:Time to Live LLDP TLV Length: 2
LLDP TLV type:LLDP Organizationally Specific LLDP TLV Length: 55
LLDP TLV type:LLDP Organizationally Specific LLDP TLV Length: 5
LLDP TLV type:END of LLDPDU LLDP TLV Length: 0
Remote Peers Information on interface Eth1/33
```

```

Remote peer's MSAP: length 12 Bytes:
00 0d ec a3 27 40 00 0d ec a3 27 68
LLDP TLV's
LLDP TLV type:Chassis ID LLDP TLV Length: 7
LLDP TLV type:Port ID LLDP TLV Length: 7
LLDP TLV type:Time to Live LLDP TLV Length: 2
LLDP TLV type:LLDP Organizationally Specific LLDP TLV Length: 55
LLDP TLV type:LLDP Organizationally Specific LLDP TLV Length: 5
LLDP TLV type:END of LLDPDU LLDP TLV Length: 0

```

This example shows how to display the system details about LLDP neighbors:

```

switch# sh lldp neighbors system-detail
Capability codes:
(R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
(W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other
Device ID Local Intf Chassis ID PortID Hold-time Capability

switch-2 Eth1/7 0005.73b7.37ce Eth1/7 120 B
switch-3 Eth/9 0005.73b7.37d0 Eth1/9 120 B
switch-4 Eth1/10 0005.73b7.37d1 Eth1/10 120 B
Total entries displayed: 3

```

This example shows how to display LLDP timer information:

```

switch# show lldp timers
LLDP Timers
holdtime 120 seconds
reinit 2 seconds
msg_tx_interval 30 seconds

```

This example shows how to display information about LLDP counters:

```

switch# show lldp traffic
LLDP traffic statistics:

Total frames out: 8464
Total Entries aged: 6
Total frames in: 6342
Total frames received in error: 2
Total frames discarded: 2
Total TLVs unrecognized: 0

```



## Configuring MAC Address Tables

---

This chapter contains the following sections:

- [Information About MAC Addresses, page 243](#)
- [Configuring MAC Addresses, page 243](#)
- [Verifying the MAC Address Configuration, page 245](#)

### Information About MAC Addresses

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

The switch dynamically builds the address table by using the MAC source address of the frames received. When the switch 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 switch adds its relevant MAC source address and port ID to the address table. The switch then forwards subsequent frames to a single LAN port without flooding all LAN ports.

You can also enter a MAC address, which is termed a static MAC address, into the table. These static MAC entries are retained across a reboot of the switch.

### Configuring MAC Addresses

#### Configuring Static MAC Addresses

You can configure static MAC addresses for the switch. These addresses can be configured in interface configuration mode or in VLAN configuration mode.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config) # **mac-address-table static** *mac\_address* **vlan** *vlan-id* {**drop** | **interface** {*type slot/port*} | **port-channel** *number*} [**auto-learn**]
3. (Optional) switch(config)# **no mac address-table static** *mac\_address* **vlan** *vlan-id*

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config) # <b>mac-address-table static</b> <i>mac_address</i> <b>vlan</b> <i>vlan-id</i> { <b>drop</b>   <b>interface</b> { <i>type slot/port</i> }   <b>port-channel</b> <i>number</i> } [ <b>auto-learn</b> ]	Specifies a static address to add to the MAC address table. If you enable the <b>auto-learn</b> option, the switch will update the entry if the same MAC address is seen on a different port.
<b>Step 3</b>	switch(config)# <b>no mac address-table static</b> <i>mac_address</i> <b>vlan</b> <i>vlan-id</i>	(Optional) Deletes the static entry from the MAC address table. Use the <b>mac address-table static</b> command to assign a static MAC address to a virtual interface.

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

```
switch# configure terminal
switch(config) # mac address-table static 12ab.47dd.ff89 vlan 3 interface ethernet 1/4
switch(config) #
```

## Configuring the Aging Time for the MAC Table

You can configure the amount of time that an entry (the packet source MAC address and port that packet ingresses) remains in the MAC table. MAC aging time can be configured in either interface configuration mode or in VLAN configuration mode.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **mac-address-table aging-time** *seconds* [**vlan** *vlan\_id*]

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>mac-address-table aging-time</b> <i>seconds</i> [ <b>vlan</b> <i>vlan_id</i> ]	Specifies the time before an entry ages out and is discarded from the MAC address table.

	Command or Action	Purpose
		The <i>seconds</i> range is from 0 to 1000000. The default is 300 seconds for Cisco NX-OS 5500 and 1800 for Cisco NX-OS 5600 and 6000 series. Entering the value 0 disables the MAC aging. If a VLAN is not specified, the aging specification applies to all VLANs.

This example shows how to set the aging time for entries in the MAC address table to 300 seconds:

```
switch# configure terminal
switch(config) # mac-address-table aging-time 300
switch(config) #
```

## Clearing Dynamic Addresses from the MAC Table

### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **clear mac-address-table dynamic** {address *mac-addr*} {interface [*type slot/port* | *port-channel number*] {vlan *vlan-id*}

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>clear mac-address-table dynamic</b> {address <i>mac-addr</i> } {interface [ <i>type slot/port</i>   <i>port-channel number</i> ] {vlan <i>vlan-id</i> }	Clears the dynamic address entries from the MAC address table.

## Verifying the MAC Address Configuration

Use one of the following commands to verify the configuration:

**Table 16: MAC Address Configuration Verification Commands**

Command	Purpose
<b>show mac-address-table aging-time</b>	Displays the MAC address aging time for all VLANs defined in the switch.

Command	Purpose
<b>show mac-address-table</b>	Displays the contents of the MAC address table. <b>Note</b> IGMP snooping learned MAC addresses are not displayed.
<b>show mac address-table loop-detect</b>	Displays the currently configured action.

This example shows how to display the MAC address table:

```
switch# show mac-address-table
VLAN      MAC Address      Type      Age      Port
-----+-----+-----+-----+-----
1         0018.b967.3cd0   dynamic   10      Eth1/3
1         001c.b05a.5380   dynamic   200     Eth1/3
Total MAC Addresses: 2
```

This example shows how to display the current aging time:

```
switch# show mac-address-table aging-time
Vlan      Aging Time
-----+-----
1         300
13        300
42        300
```

This example shows how to display the currently configured action:

```
switch# configure terminal
switch(config)# show mac address-table loop-detect
Port Down Action Mac Loop Detect : enabled
```

```
switch# configure terminal
switch(config)# no mac address-table loop-detect port-down
switch(config)# show mac address-table loop-detect
Port Down Action Mac Loop Detect : disabled
```



## Configuring IGMP Snooping

---

This chapter contains the following sections:

- [Information About IGMP Snooping, page 247](#)
- [Configuring IGMP Snooping Parameters, page 250](#)
- [Verifying the IGMP Snooping Configuration, page 253](#)

### Information About IGMP Snooping

The IGMP snooping software examines IGMP protocol messages within a VLAN to discover which interfaces are connected to hosts or other devices interested in receiving this traffic. Using the interface information, IGMP snooping can reduce bandwidth consumption in a multiaccess LAN environment to avoid flooding the entire VLAN. The IGMP snooping feature tracks which ports are attached to multicast-capable routers to help it manage the forwarding of IGMP membership reports. The IGMP snooping software responds to topology change notifications.



**Note**

---

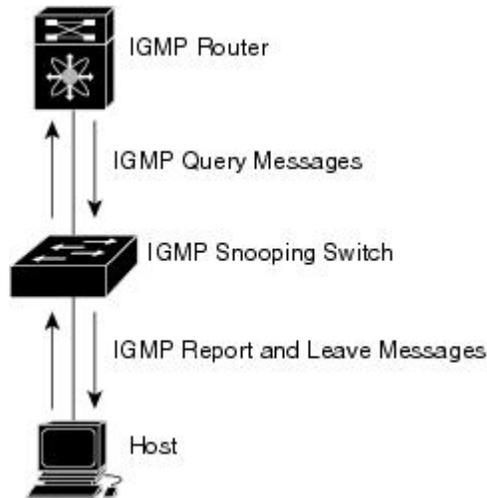
IGMP snooping is supported on all Ethernet interfaces. The term *snooping* is used because Layer 3 control plane packets are intercepted and influence Layer 2 forwarding decisions.

---

Cisco NX-OS supports IGMPv2 and IGMPv3. IGMPv2 supports IGMPv1, and IGMPv3 supports IGMPv2. Although not all features of an earlier version of IGMP are supported, the features related to membership query and membership report messages are supported for all IGMP versions.

The following figure shows an IGMP snooping switch that is located between the host and the IGMP router. The IGMP snooping switch snoops the IGMP membership reports and leave messages and forwards them only when necessary to the connected IGMP routers.

**Figure 26: IGMP Snooping Switch**



**Note** The switch supports IGMPv3 snooping based only on the destination multicast MAC address. It does not support snooping based on the source MAC address or on proxy reports.

The Cisco NX-OS IGMP snooping software supports optimized multicast flooding (OMF) that forwards unknown traffic to routers only and performs no data driven state creation. For more information about IGMP snooping, see <http://tools.ietf.org/wg/magma/draft-ietf-magma-snoop/rfc4541.txt>.

## IGMPv1 and IGMPv2

Both IGMPv1 and IGMPv2 support membership report suppression, which means that if two hosts on the same subnet want to receive multicast data for the same group, the host that receives a member report from the other host suppresses sending its report. Membership report suppression occurs for hosts that share a port.

If no more than one host is attached to each VLAN switch port, you can configure the fast leave feature in IGMPv2. The fast leave feature does not send last member query messages to hosts. As soon as the software receives an IGMP leave message, the software stops forwarding multicast data to that port.

IGMPv1 does not provide an explicit IGMP leave message, so the software must rely on the membership message timeout to indicate that no hosts remain that want to receive multicast data for a particular group.



**Note** Cisco NX-OS ignores the configuration of the last member query interval when you enable the fast leave feature because it does not check for remaining hosts.

## IGMPv3

The IGMPv3 snooping implementation on the switch forwards IGMPv3 reports to allow the upstream multicast router to do source-based filtering.

By default, the software tracks hosts on each VLAN port. The explicit tracking feature provides a fast leave mechanism. Because every IGMPv3 host sends membership reports, a report suppression feature limits the amount of traffic the switch sends to other multicast-capable routers. When report suppression is enabled, and no IGMPv1 or IGMPv2 hosts request the same group, the software provides proxy reporting. The proxy feature builds the group state from membership reports from the downstream hosts and generates membership reports in response to queries from upstream queriers.

Even though the IGMPv3 membership reports provide a full accounting of group members on a LAN segment, when the last host leaves, the software sends a membership query. You can configure the parameter last member query interval. If no host responds before the timeout, the software removes the group state.

## IGMP Snooping Querier

When there is no multicast router in the VLAN to originate the queries, you must configure an IGMP snooping querier to send membership queries.

When an IGMP snooping querier is enabled, it sends out periodic IGMP queries that trigger IGMP report messages from hosts that want to receive IP multicast traffic. IGMP snooping listens to these IGMP reports to establish appropriate forwarding.

Currently, you can configure the same SVI IP address for the switch querier and the IGMP snooping querier. Both queriers will then be active at the same time, and both queriers will send general queries to the VLAN periodically. To prevent this from happening, ensure that you use different IP addresses for the IGMP snooping querier and the switch querier.

## IGMP Forwarding

The control plane of the Cisco Nexus device is able to detect IP addresses but forwarding occurs using the MAC address only.

When a host connected to the switch wants to join an IP multicast group, it sends an unsolicited IGMP join message, specifying the IP multicast group to join. Alternatively, when the switch receives a general query from a connected router, it forwards the query to all interfaces, physical and virtual, in the VLAN. Hosts wanting to join the multicast group respond by sending a join message to the switch. The switch CPU creates a multicast forwarding table entry for the group if it is not already present. The CPU also adds the interface where the join message was received to the forwarding table entry. The host associated with that interface receives multicast traffic for that multicast group.

The router sends periodic multicast general queries and the switch forwards these queries through all ports in the VLAN. Interested hosts respond to the queries. If at least one host in the VLAN wants to receive multicast traffic, the router continues forwarding the multicast traffic to the VLAN. The switch forwards multicast group traffic to only those hosts listed in the forwarding table for that multicast group.

When hosts want to leave a multicast group, they can either silently leave, or they can send a leave message. When the switch receives a leave message from a host, it sends a group-specific query to determine if any other devices connected to that interface are interested in traffic for the specific multicast group. The switch then updates the forwarding table for that MAC group so that only those hosts interested in receiving multicast

traffic for the group are listed in the forwarding table. If the router receives no reports from a VLAN, it removes the group for the VLAN from its IGMP cache.

## Configuring IGMP Snooping Parameters

To manage the operation of the IGMP snooping process, you can configure the optional IGMP snooping parameters described in the following table.

**Table 17: IGMP Snooping Parameters**

Parameter	Description
IGMP snooping	Enables IGMP snooping on a per-VLAN basis. The default is enabled.  <b>Note</b> If the global setting is disabled, all VLANs are treated as disabled, whether they are enabled or not.
Explicit tracking	Tracks IGMPv3 membership reports from individual hosts for each port on a per-VLAN basis. The default is enabled.
Fast leave	Enables the software to remove the group state when it receives an IGMP Leave report without sending an IGMP query message. This parameter is used for IGMPv2 hosts when no more than one host is present on each VLAN port. The default is disabled.
Last member query interval	Sets the interval that the software waits after sending an IGMP query to verify that no hosts that want to receive a particular multicast group remain on a network segment. If no hosts respond before the last member query interval expires, the software removes the group from the associated VLAN port. Values range from 1 to 25 seconds. The default is 1 second.
Snooping querier	Configures a snooping querier on an interface when there is no multicast router in the VLAN to generate queries. The default is disabled.
Report suppression	Limits the membership report traffic sent to multicast-capable routers. When you disable report suppression, all IGMP reports are sent as is to multicast-capable routers. The default is enabled.
Multicast router	Configures a static connection to a multicast router. The interface to the router must be in the selected VLAN.

Parameter	Description
Multicast router vpc-peer-link	<p>Configures a static connection to a virtual port channel (vPC) peer link.</p> <p>By default, the vPC peer link is considered a multicast router port and the multicast packet is sent to the peer link for each receiver VLAN.</p> <p>To send the multicast traffic over a vPC peer link to each receiver VLAN that has orphan ports, use the <b>no ip igmp snooping mrouter vpc-peer-link</b> command. If you use the <b>no ip igmp snooping mrouter vpc-peer-link</b> command, the multicast traffic is not sent over to a peer link for the source VLAN and receiver VLAN unless there is an orphan port in the VLAN. The IGMP snooping mrouter VPC peer link should also be globally disabled on the peer VPC switch.</p>
Static group	Configures an interface that belongs to a VLAN as a static member of a multicast group.

You can disable IGMP snooping either globally or for a specific VLAN.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **ip igmp snooping**
3. switch(config)# **vlan configuration** *vlan-id*
4. switch(config-vlan)# **ip igmp snooping**
5. switch(config-vlan)# **ip igmp snooping explicit-tracking**
6. switch(config-vlan)# **ip igmp snooping fast-leave**
7. switch(config-vlan)# **ip igmp snooping last-member-query-interval** *seconds*
8. switch(config-vlan)# **ip igmp snooping querier** *IP-address*
9. switch(config-vlan)# **ip igmp snooping report-suppression**
10. switch(config-vlan)# **ip igmp snooping mrouter interface** *interface*
11. switch(config-vlan)# **ip igmp snooping mrouter vpc-peer-link**
12. switch(config-vlan)# **ip igmp snooping static-group** *group-ip-addr* [*source source-ip-addr*] **interface** *interface*

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.

	Command or Action	Purpose
<b>Step 2</b>	switch(config)# <b>ip igmp snooping</b>	Globally enables IGMP snooping. The default is enabled. <b>Note</b> If the global setting is disabled, all VLANs are treated as disabled, whether they are enabled or not.
<b>Step 3</b>	switch(config)# <b>vlan configuration</b> <i>vlan-id</i>	Enters VLAN configuration mode.
<b>Step 4</b>	switch(config-vlan)# <b>ip igmp snooping</b>	Enables IGMP snooping for the current VLAN. The default is enabled. <b>Note</b> If IGMP snooping is enabled globally, this command is not required.
<b>Step 5</b>	switch(config-vlan)# <b>ip igmp snooping explicit-tracking</b>	Tracks IGMPv3 membership reports from individual hosts for each port on a per-VLAN basis. The default is enabled on all VLANs.
<b>Step 6</b>	switch(config-vlan)# <b>ip igmp snooping fast-leave</b>	Supports IGMPv2 hosts that cannot be explicitly tracked because of the host report suppression mechanism of the IGMPv2 protocol. When you enable fast leave, the IGMP software assumes that no more than one host is present on each VLAN port. The default is disabled for all VLANs.
<b>Step 7</b>	switch(config-vlan)# <b>ip igmp snooping last-member-query-interval</b> <i>seconds</i>	Removes the group from the associated VLAN port if no hosts respond to an IGMP query message before the last member query interval expires. Values range from 1 to 25 seconds. The default is 1 second.
<b>Step 8</b>	switch(config-vlan)# <b>ip igmp snooping querier</b> <i>IP-address</i>	Configures a snooping querier when you do not enable PIM because multicast traffic does not need to be routed. The IP address is used as the source in messages. The default is disabled.
<b>Step 9</b>	switch(config-vlan)# <b>ip igmp snooping report-suppression</b>	Limits the membership report traffic sent to multicast-capable routers. When you disable report suppression, all IGMP reports are sent as is to multicast-capable routers. The default is enabled.
<b>Step 10</b>	switch(config-vlan)# <b>ip igmp snooping mrouter interface</b> <i>interface</i>	Configures a static connection to a multicast router. The interface to the router must be in the selected VLAN. You can specify the interface by type and number.
<b>Step 11</b>	switch(config-vlan)# <b>ip igmp snooping mrouter vpc-peer-link</b>	Configures a static connection to a virtual port channel (vPC) peer link. By default, the vPC peer link is considered as a multicast router port and the multicast packet is sent to the peer link for each receiver VLAN. To send the multicast traffic over a vPC peer link to each receiver VLAN that has orphan ports, use the <b>no ip igmp snooping mrouter vpc-peer-link</b> command. The IGMP snooping mrouter VPC peer link should also be globally disabled on the peer VPC switch.
<b>Step 12</b>	switch(config-vlan)# <b>ip igmp snooping static-group</b> <i>group-ip-addr</i> [ <b>source</b> <i>source-ip-addr</i> ] <b>interface</b> <i>interface</i>	Configures an interface belonging to a VLAN as a static member of a multicast group. You can specify the interface by type and number.

This example shows how to configure IGMP snooping parameters for a VLAN:

```
switch# configure terminal
switch(config)# vlan configuration 5
switch(config-vlan)# ip igmp snooping last-member-query-interval 3
switch(config-vlan)# ip igmp snooping querier 172.20.52.106
switch(config-vlan)# ip igmp snooping explicit-tracking
switch(config-vlan)# ip igmp snooping fast-leave
switch(config-vlan)# ip igmp snooping report-suppression
switch(config-vlan)# ip igmp snooping mrouter interface ethernet 1/10
switch(config-vlan)# ip igmp snooping mrouter vpc-peer-link
switch(config-vlan)# ip igmp snooping static-group 230.0.0.1 interface ethernet 1/10
switch(config-vlan)# end
```

This example shows how to configure a static connection to a vPC peer link and how to remove the static connection to a vPC peer link:

```
switch(config)# ip igmp snooping mrouter vpc-peer-link
switch(config)# no ip igmp snooping mrouter vpc-peer-link
Warning: IGMP Snooping mrouter vpc-peer-link should be globally disabled on peer VPC switch
as well.
switch(config)#
```

## Verifying the IGMP Snooping Configuration

Use the following commands to verify the IGMP snooping configuration.

Command	Description
<code>show ip igmp snooping [[vlan] vlan-id]</code>	Displays IGMP snooping configuration by VLAN.
<code>show ip igmp snooping groups [[vlan] vlan-id] [detail]</code>	Displays IGMP snooping information about groups by VLAN.
<code>show ip igmp snooping querier [[vlan] vlan-id]</code>	Displays IGMP snooping queriers by VLAN.
<code>show ip igmp snooping mrouter [[vlan] vlan-id]</code>	Displays multicast router ports by VLAN.
<code>show ip igmp snooping explicit-tracking vlan vlan-id</code>	Displays IGMP snooping explicit tracking information by VLAN.

This example shows how to verify the IGMP snooping parameters:

```
switch# show ip igmp snooping
Global IGMP Snooping Information:
  IGMP Snooping enabled
IGMP Snooping information for vlan 1
  IGMP snooping enabled
  IGMP querier none
  Switch-querier disabled
  Explicit tracking enabled
  Fast leave disabled
  Report suppression enabled
  Router port detection using PIM Hellos, IGMP Queries
  Number of router-ports: 0
  Number of groups: 0
IGMP Snooping information for vlan 5
IGMP snooping enabled
  IGMP querier present, address: 192.0.2.1, version: 3
  Querier interval: 125 secs
```

```
Querier last member query interval: 10 secs
Querier robustness: 2
Switch-querier enabled, address 192.0.2.1, currently running
Explicit tracking enabled
Fast leave enabled
Report suppression enabled
Router port detection using PIM Hellos, IGMP Queries
Number of router-ports: 1
Number of groups: 1
```



## Configuring MVR

---

This chapter contains the following sections:

- [Information About MVR, page 255](#)
- [Licensing Requirements for MVR, page 256](#)
- [Guidelines and Limitations for MVR, page 256](#)
- [Default MVR Settings, page 257](#)
- [Configuring MVR, page 257](#)
- [Verifying the MVR Configuration, page 260](#)

## Information About MVR

### MVR Overview

In a typical Layer 2 multi-VLAN network, subscribers to a multicast group can be on multiple VLANs. To maintain data isolation between these VLANs, the multicast stream on the source VLAN must be passed to a router, which replicates the stream on all subscriber VLANs, wasting upstream bandwidth.

Multicast VLAN Registration (MVR) allows a Layer 2 switch to forward the multicast data from a source on a common assigned VLAN to the subscriber VLANs, conserving upstream bandwidth by bypassing the router. The switch forwards multicast data for MVR IP multicast streams only to MVR ports on which hosts have joined, either by IGMP reports or by MVR static configuration. The switch forwards IGMP reports received from MVR hosts only to the source port. For other traffic, VLAN isolation is preserved.

MVR requires at least one VLAN to be designated as the common VLAN to carry the multicast stream from the source. More than one such multicast VLAN (MVR VLAN) can be configured in the system, and you can configure a global default MVR VLAN as well as interface-specific default MVR VLANs. Each multicast group using MVR is assigned to an MVR VLAN.

MVR allows a subscriber on a port to subscribe and unsubscribe to a multicast stream on the MVR VLAN by sending IGMP join and leave messages. IGMP leave messages from an MVR group are handled according to the IGMP configuration of the VLAN on which the leave message is received. If IGMP fast leave is enabled

on the VLAN, the port is removed immediately; otherwise, an IGMP query is sent to the group to determine whether other hosts are present on the port.

## MVR Interoperation with Other Features

### MVR and IGMP Snooping

Although MVR operates on the underlying mechanism of IGMP snooping, the two features operate independently of each other. One feature can be enabled or disabled without affecting the operation of the other feature. If IGMP snooping is disabled globally or on a VLAN, and if MVR is enabled on the VLAN, IGMP snooping is internally enabled on the VLAN. Joins received for MVR groups on non-MVR receiver ports, or joins received for non-MVR groups on MVR receiver ports, are processed by IGMP snooping.

### MVR and vPC

- As with IGMP snooping, IGMP control messages received by virtual port channel (vPC) peer switches are exchanged between the peers, allowing synchronization of MVR group information.
- MVR configuration must be consistent between the peers.
- The **no ip igmp snooping mrouter vpc-peer-link** command applies to MVR. With this command, multicast traffic is not sent over to a peer link for the source VLAN and receiver VLAN unless there is an orphan port in the VLAN.
- The **show mvr member** command shows the multicast group on the vPC peer switch. However, the vPC peer switch does not show the multicast groups if it does not receive the IGMP membership report of the groups.

## Licensing Requirements for MVR

The following table shows the licensing requirements for this feature:

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

## Guidelines and Limitations for MVR

When configuring MVR, follow these guidelines:

- MVR is supported only on Layer 2 Ethernet ports, such as individual ports, port channels, and virtual Ethernet (vEth) ports.
- MVR receiver ports can only be access ports; they cannot be trunk ports. MVR source ports can be either access or trunk ports.

- MVR configuration on Flex Link ports is not supported.
- Priority tagging is not supported on MVR receiver ports.
- When using private VLANs (PVLANS), you cannot configure a secondary VLAN as the MVR VLAN.
- The total number of MVR VLANs cannot exceed 250.

**Note**

During and in-service software upgrade (ISSU), MVR IGMP membership for the MVR receiver ports may timeout because the joins are not forwarded to the upstream router. In order to avoid a timeout, the querier timer on the upstream router or the network querier should be increased to accommodate an ISSU.

## Default MVR Settings

Parameter	Default
MVR	Disabled globally and per interface
Global MVR VLAN	None configured
Interface (per port) default	Neither a receiver nor a source port

## Configuring MVR

### Configuring MVR Global Parameters

#### SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# [**no**] **mvr**
3. switch(config)# [**no**] **mvr-vlan** *vlan-id*
4. switch(config)# [**no**] **mvr-group** *addr[/mask]* [**count** *groups*] [**vlan** *vlan-id*]
5. (Optional) switch(config)# **end**
6. (Optional) switch# **clear mvr counters** [**source-ports** | **receiver-ports**]
7. (Optional) switch# **show mvr**
8. (Optional) switch# **copy running-config startup-config**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# <b>configure terminal</b>	Enters global configuration mode.
Step 2	switch(config)# <b>[no] mvr</b>	Globally enables MVR. The default is disabled. Use the <b>no</b> form of the command to disable MVR.
Step 3	switch(config)# <b>[no] mvr-vlan vlan-id</b>	Specifies the global default MVR VLAN. The MVR VLAN is the source of the multicast message that subsequent receivers subscribe to. The range is 1 to 4094. Use the <b>no</b> form of the command to clear the MVR VLAN.
Step 4	switch(config)# <b>[no] mvr-group addr[/mask] [count groups] [vlan vlan-id]</b>	Adds a multicast group at the specified IPv4 address and (optional) netmask length to the global default MVR VLAN. You can repeat this command to add additional groups to the MVR VLAN. The IP address is entered in the format <i>a.b.c.d/m</i> , where <i>m</i> is the number of bits in the netmask, from 1 to 31. (Optional) You can specify a number of MVR groups using contiguous multicast IP addresses starting with the specified IP address. Use the <b>count</b> keyword followed by a number from 1 to 64. (Optional) You can explicitly specify an MVR VLAN for the group by using the <b>vlan</b> keyword; otherwise, the group is assigned to the default MVR VLAN. Use the <b>no</b> form of the command to clear the group configuration.
Step 5	switch(config)# <b>end</b>	(Optional) Returns to privileged EXEC mode.
Step 6	switch# <b>clear mvr counters [source-ports   receiver-ports]</b>	(Optional) Clears MVR IGMP packet counters.
Step 7	switch# <b>show mvr</b>	(Optional) Displays the global MVR configuration.
Step 8	switch# <b>copy running-config startup-config</b>	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to globally enable MVR and configure the global parameters:

```
switch# configure terminal
switch(config)# mvr
switch(config-mvr)# mvr-vlan 100
switch(config-mvr)# mvr-group 192.0.2.1 count 4
switch(config-mvr)# mvr-group 192.0.2.240/28 vlan 101
switch(config-mvr)# mvr-group 192.0.2.6 vlan 340
switch(config-mvr)# end
switch# show mvr
MVR Status           : enabled
Global MVR VLAN      : 100
```

```
Number of MVR VLANs : 3
switch# copy running-config startup-config
```

## Configuring MVR Interfaces

### SUMMARY STEPS

1. switch# **configure terminal**
2. **mvr**
3. **interface** {*ethernet type slot/port* | **port-channel** *channel-number* | **vethernet** *number*}
4. [**no**] **mvr-type** {**source** | **receiver**}
5. (Optional) [**no**] **mvr-vlan** *vlan-id*
6. (Optional) [**no**] **mvr-group** *addr[/mask]* [**vlan** *vlan-id*]
7. (Optional) **end**
8. (Optional) switch# **copy running-config startup-config**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	<b>mvr</b>	Globally enables MVR. The default is disabled.  <b>Note</b> If MVR is enabled globally, then this command is not required.
<b>Step 3</b>	<b>interface</b> { <i>ethernet type slot/port</i>   <b>port-channel</b> <i>channel-number</i>   <b>vethernet</b> <i>number</i> }	Specifies the Layer 2 port to configure, and enters interface configuration mode.
<b>Step 4</b>	[ <b>no</b> ] <b>mvr-type</b> { <b>source</b>   <b>receiver</b> }	Configures an MVR port as one of these types of ports: <ul style="list-style-type: none"> <li>• <b>source</b>—An uplink port that sends and receives multicast data is configured as an MVR source. The port automatically becomes a static receiver of MVR multicast groups. A source port should be a member of the MVR VLAN.</li> <li>• <b>receiver</b>— An access port that is connected to a host that wants to subscribe to an MVR multicast group is configured as an MVR receiver. A receiver port receives data only when it becomes a member of the multicast group by using IGMP leave and join messages.</li> </ul> <p>If you attempt to configure a non-MVR port with MVR characteristics, the configuration is cached and does not take effect until the port becomes an MVR port. The default port mode is non-MVR.</p>

	Command or Action	Purpose
Step 5	<code>[no] mvr-vlan vlan-id</code>	(Optional) Specifies an interface default MVR VLAN that overrides the global default MVR VLAN for joins received on the interface. The MVR VLAN is the source of the multicast message that subsequent receivers subscribe to.  The range is 1 to 4094.
Step 6	<code>[no] mvr-group addr[/mask]</code> <code>[vlan vlan-id]</code>	(Optional) Adds a multicast group at the specified IPv4 address and (optional) netmask length to the interface MVR VLAN, overriding the global MVR group configuration. You can repeat this command to add additional groups to the MVR VLAN  The IP address is entered in the format <i>a.b.c.d/m</i> , where <i>m</i> is the number of bits in the netmask, from 1 to 31.  (Optional) You can explicitly specify an MVR VLAN for the group by using the <b>vlan</b> keyword; otherwise, the group is assigned to the interface default (if specified) or global default MVR VLAN.  Use the <b>no</b> form of the command to clear the IPv4 address and netmask.
Step 7	<code>end</code>	(Optional) Return to privileged EXEC mode.
Step 8	<code>switch# copy running-config startup-config</code>	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

This example shows how to configure an Ethernet port as an MVR receiver port:

```
switch# configure terminal
switch(config)# mvr
switch(config-mvr)# interface ethernet 1/10
switch(config-if)# mvr-type receiver
switch(config-if)# end
switch# copy running-config startup-config
switch#
```

## Verifying the MVR Configuration

Use the following commands to verify the MVR configuration:

Command	Description
<code>show mvr</code>	Displays the MVR subsystem configuration and status.
<code>show mvr groups</code>	Displays the MVR group configuration.

Command	Description
<b>show ip igmp snooping</b> [ <b>vlan</b> <i>vlan-id</i> ]	Displays information about IGMP snooping on the specified VLAN.
<b>show mvr interface</b> { <b>ethernet</b> <i>type slot/port</i>   <b>port-channel</b> <i>number</i> }	Displays the MVR configuration on the specified interface.
<b>show mvr members</b> [ <b>count</b> ]	Displays the number and details of all MVR receiver members.
<b>show mvr members interface</b> { <b>ethernet</b> <i>type slot/port</i>   <b>port-channel</b> <i>number</i> }	Displays details of MVR members on the specified interface.
<b>show mvr members vlan</b> <i>vlan-id</i>	Displays details of MVR members on the specified VLAN.
<b>show mvr receiver-ports</b> [ <b>ethernet</b> <i>type slot/port</i>   <b>port-channel</b> <i>number</i> ]	Displays all MVR receiver ports on all interfaces or on the specified interface.
<b>show mvr source-ports</b> [ <b>ethernet</b> <i>type slot/port</i>   <b>port-channel</b> <i>number</i> ]	Displays all MVR source ports on all interfaces or on the specified interface.

This example shows how to verify the MVR parameters:

```
switch# show mvr
MVR Status      : enabled
Global MVR VLAN : 100
Number of MVR VLANs : 4
```

This example shows how to verify the MVR group configuration:

```
switch# show mvr groups
* - Global default MVR VLAN.

Group start      Group end      Count  MVR-VLAN  Interface
                Mask
-----
228.1.2.240     228.1.2.255   /28    101
230.1.1.1       230.1.1.4     4      *100
235.1.1.6       235.1.1.6     1      340
225.1.3.1       225.1.3.1     1      *100      Eth1/10
```

This example shows how to verify the MVR interface configuration and status:

```
switch# show mvr interface
Port      VLAN  Type      Status      MVR-VLAN
-----
Po10      100   SOURCE    ACTIVE      100-101
Po201     201   RECEIVER  ACTIVE      100-101,340
Po202     202   RECEIVER  ACTIVE      100-101,340
Po203     203   RECEIVER  ACTIVE      100-101,340
Po204     204   RECEIVER  INACTIVE    100-101,340
Po205     205   RECEIVER  ACTIVE      100-101,340
Po206     206   RECEIVER  ACTIVE      100-101,340
Po207     207   RECEIVER  ACTIVE      100-101,340
Po208     208   RECEIVER  ACTIVE      2000-2001
Eth1/9    340   SOURCE    ACTIVE      340
Eth1/10   20    RECEIVER  ACTIVE      100-101,340
Eth2/2    20    RECEIVER  ACTIVE      100-101,340
Eth102/1/1 102   RECEIVER  ACTIVE      100-101,340
```

```
Eth102/1/2 102 RECEIVER INACTIVE 100-101,340
Eth103/1/1 103 RECEIVER ACTIVE 100-101,340
Eth103/1/2 103 RECEIVER ACTIVE 100-101,340
```

Status INVALID indicates one of the following misconfiguration:

- a) Interface is not a switchport.
- b) MVR receiver is not in access, pvlan host or pvlan promiscuous mode.
- c) MVR source is in fex-fabric mode.

This example shows how to display all MVR members:

```
switch# show mvr members
MVR-VLAN  Group Address  Status  Members
-----
100       230.1.1.1  ACTIVE  Po201 Po202 Po203 Po205 Po206
100       230.1.1.2  ACTIVE  Po205 Po206 Po207 Po208
340       235.1.1.6  ACTIVE  Eth102/1/1
101       225.1.3.1  ACTIVE  Eth1/10 Eth2/2
101       228.1.2.241 ACTIVE  Eth103/1/1 Eth103/1/2
```

This example shows how to display all MVR receiver ports on all interfaces:

```
switch# show mvr receiver-ports
Port          MVR-VLAN  Status  Joins      Leaves
              (v1,v2,v3)
-----
Po201         100       ACTIVE  8          2
Po202         100       ACTIVE  8          2
Po203         100       ACTIVE  8          2
Po204         100       INACTIVE 0          0
Po205         100       ACTIVE  10         6
Po206         100       ACTIVE  10         6
Po207         100       ACTIVE  5          0
Po208         100       ACTIVE  6          0
Eth1/10       101       ACTIVE  12         2
Eth2/2        101       ACTIVE  12         2
Eth102/1/1    340       ACTIVE  16         15
Eth102/1/2    340       INACTIVE 16         16
Eth103/1/1    101       ACTIVE  33         0
Eth103/1/2    101       ACTIVE  33         0
```

This example shows how to display all MVR source ports on all interfaces:

```
switch# show mvr source-ports
Port          MVR-VLAN  Status
-----
Po10          100       ACTIVE
Eth1/9        340       ACTIVE
```



## Configuring Traffic Storm Control

---

This chapter contains the following sections:

- [Information About Traffic Storm Control, page 263](#)
- [Guidelines and Limitations for Traffic Storm Control, page 265](#)
- [Configuring Traffic Storm Control, page 266](#)
- [Traffic Storm Control Example Configuration, page 267](#)
- [Default Settings for Traffic Storm Control, page 267](#)

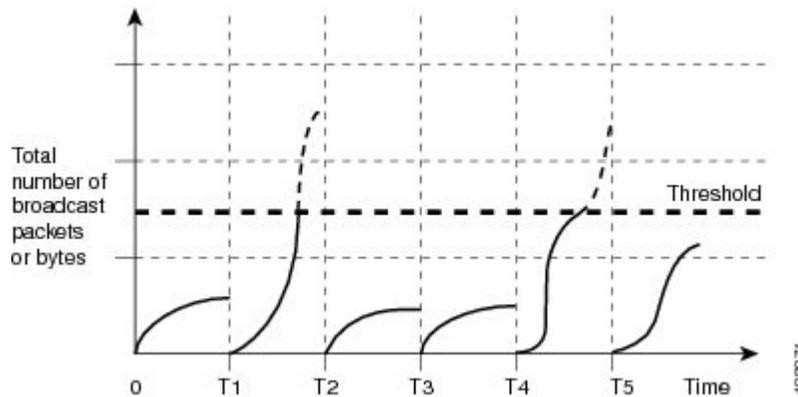
### Information About Traffic Storm Control

A traffic storm occurs when packets flood the LAN, creating excessive traffic and degrading network performance. You can use the traffic storm control feature to prevent disruptions on Ethernet interfaces by a broadcast, multicast, or unknown unicast traffic storm.

Traffic storm control (also called traffic suppression) allows you to monitor the levels of the incoming broadcast, multicast, or unknown unicast traffic over a 10-microsecond 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.

The following figure shows the broadcast traffic patterns on an Ethernet interface during a specified time interval. In this example, traffic storm control occurs between times T1 and T2 and between T4 and T5. During those intervals, the amount of broadcast traffic exceeded the configured threshold.

**Figure 27: Broadcast Suppression**



The traffic storm control threshold numbers and the time interval allow the traffic storm control algorithm to work with different levels of packet granularity. For example, a higher threshold allows more packets to pass through.

Traffic storm control is implemented in the hardware. The traffic storm control circuitry monitors packets that pass from an Ethernet interface to the switching bus. Using the Individual/Group bit in the packet destination address, the circuitry determines if the packet is unicast or broadcast, tracks the current count of packets within the 10-microsecond interval, and filters out subsequent packets when a threshold is reached.

Traffic storm control uses a bandwidth-based method to measure traffic. You set the percentage of total available bandwidth that the controlled traffic can use. Because packets do not arrive at uniform intervals, the 10-microsecond interval can affect the operation of traffic storm control.

The following are examples of how traffic storm control operation is affected:

- If you enable broadcast traffic storm control, and broadcast traffic exceeds the level within the 10-microsecond interval, traffic storm control drops all exceeding broadcast traffic until the end of the interval.
- If you enable multicast traffic storm control, and the multicast traffic exceeds the level within the 10-microsecond interval, traffic storm control drops all exceeding multicast traffic until the end of the interval.
- If you enable broadcast and multicast traffic storm control, and broadcast traffic exceeds the level within the 10-microsecond interval, traffic storm control drops all exceeding broadcast traffic until the end of the interval.
- If you enable broadcast and multicast traffic storm control, and multicast traffic exceeds the level within the 10-microsecond interval, traffic storm control drops all exceeding multicast traffic until the end of the interval.

By default, Cisco NX-OS takes no corrective action when traffic exceeds the configured level.

# Guidelines and Limitations for Traffic Storm Control

When configuring the traffic storm control level, follow these guidelines and limitations:

- You can configure traffic storm control on a port-channel interface.
- You can configure traffic storm control on a fabric port or a fabric port channel that connects the switch to a Fabric Extender (FEX). Storm control configured on a FEX applies to the aggregate traffic coming in on all the ports on that FEX.

**Note**

The NIF storm control feature applies on all traffic coming in on a FEX fabric port. Traffic that comes on the FEX fabric port with the VNTAG header has an additional 6 bytes added to the original traffic. Due to these additional 6 bytes of overhead, the rate at which the traffic is policed by the storm control policer is skewed depending on the packet size of the original traffic that is ingressing on the HIF ports. The skew is larger for the smaller packet sizes compared to the larger packet sizes.

- Specify the level as a percentage of the total interface bandwidth:
  - The level can be from 0 to 100.
  - The optional fraction of a level can be from 0 to 99.
  - 100 percent means no traffic storm control.
  - 0.0 percent suppresses all traffic.
- There are local link and hardware limitations that prevent storm-control drops from being counted separately. Instead, storm-control drops are counted with other drops in the discards counter.
- Because of hardware limitations and the method by which packets of different sizes are counted, the level percentage is an approximation. Depending on the sizes of the frames that make up the incoming traffic, the actual enforced level might differ from the configured level by several percentage points.
- Applying storm control over a HIF range is not recommended. The configuration might fail for one or more interfaces in the range depending on the hardware resource availability. The result of the command is partial success in some cases.
- In the Cisco Nexus 5000 switch, storm-control does not distinguish between IP, non-IP, registered, or unregistered multicast traffic. All multicast traffic is subject to a single-multicast storm control policer when configured.
- In the Cisco Nexus 5500 switch, storm-control is applied only to unregistered or unknown multicast MAC address.
- The link-level control protocols (LACP, LLDP and so on) are not affected in case of a traffic storm. The storm control is applied to data plane traffic only.
- The burst size values are:
  - For a 10G port, 48.68 Mbytes/390Mbits
  - For a 1G port, 25 Mbytes/200Mbits

# Configuring Traffic Storm Control

You can set the percentage of total available bandwidth that the controlled traffic can use.


**Note**

Traffic storm control uses a 10-microsecond interval that can affect the operation of traffic storm control.

## SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** {**ethernet slot/port** | **port-channel number**}
3. switch(config-if)# **storm-control** [**broadcast** | **multicast** | **unicast**] **level percentage** [*fraction*]

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	switch# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 2</b>	switch(config)# <b>interface</b> { <b>ethernet slot/port</b>   <b>port-channel number</b> }	Enters interface configuration mode.
<b>Step 3</b>	switch(config-if)# <b>storm-control</b> [ <b>broadcast</b>   <b>multicast</b>   <b>unicast</b> ] <b>level percentage</b> [ <i>fraction</i> ]	Configures traffic storm control for traffic on the interface. The default state is disabled.

This example shows how to configure traffic storm control for port channels 122 and 123:

```
switch# configure terminal
switch(config)# interface port-channel 122, port-channel 123
switch(config-if-range)# storm-control unicast level 66.75
switch(config-if-range)# storm-control multicast level 66.75
switch(config-if-range)# storm-control broadcast level 66.75
switch(config-if-range)#
```

## Verifying the Traffic Storm Control Configuration

Use the following commands to display traffic storm control configuration information:

Command	Purpose
<b>show interface</b> [ <b>ethernet slot/port</b>   <b>port-channel number</b> ] <b>counters storm-control</b>	Displays the traffic storm control configuration for the interfaces.  <b>Note</b> Traffic storm control uses a 10-microsecond interval that can affect the operation of traffic storm control.
<b>show running-config interface</b>	Displays the traffic storm control configuration.

**Note**

When a storm event occurs on a port and the packets are dropped due to storm control configuration, a syslog message is generated to indicate that the storm event has started. An additional syslog message is generated when the storm event ends and the packet are no longer dropped.

## Traffic Storm Control Example Configuration

This example shows how to configure traffic storm control:

## Default Settings for Traffic Storm Control

The following table lists the default settings for traffic storm control parameters.

**Table 18: Default Traffic Storm Control Parameters**

Parameters	Default
Traffic storm control	Disabled
Threshold percentage	100





## Configuring the Fabric Extender

---

This chapter contains the following sections:

- [Information About the Cisco Nexus 2000 Series Fabric Extender, page 269](#)
- [Fabric Extender Terminology, page 270](#)
- [Fabric Extender Features, page 271](#)
- [Oversubscription, page 274](#)
- [Management Model, page 274](#)
- [Forwarding Model, page 275](#)
- [Connection Model, page 275](#)
- [Port Numbering Convention, page 278](#)
- [Fabric Extender Image Management, page 278](#)
- [Fabric Extender Hardware, page 278](#)
- [Associating a Fabric Extender to a Fabric Interface, page 279](#)
- [Configuring Fabric Extender Global Features, page 283](#)
- [Enabling the Fabric Extender Locator LED, page 285](#)
- [Redistributing the Links, page 285](#)
- [Verifying the Fabric Extender Configuration, page 287](#)
- [Verifying the Chassis Management Information, page 290](#)
- [Configuring the Cisco Nexus N2248TP-E Fabric Extender, page 294](#)

### Information About the Cisco Nexus 2000 Series Fabric Extender

The Cisco Nexus 2000 Series Fabric Extender, also known as FEX, is a highly scalable and flexible server networking solution that works with Cisco Nexus Series devices to provide high-density, low-cost connectivity for server aggregation. Scaling across 1-Gigabit Ethernet, 10-Gigabit Ethernet, unified fabric, rack, and blade server environments, the Fabric Extender is designed to simplify data center architecture and operations.

The Fabric Extender integrates with its parent switch, which is a Cisco Nexus Series device, to allow automatic provisioning and configuration taken from the settings on the parent device. This integration allows large numbers of servers and hosts to be supported by using the same feature set as the parent device with a single management domain. The Fabric Extender and its parent switch enable a large multipath, loop-free, active-active data center topology without the use of the Spanning Tree Protocol (STP).

The Cisco Nexus 2000 Series Fabric Extender forwards all traffic to its parent Cisco Nexus Series device over 10-Gigabit Ethernet fabric uplinks, which allows all traffic to be inspected by policies established on the Cisco Nexus Series device.

No software is included with the Fabric Extender. The software is automatically downloaded and upgraded from its parent device.

## Fabric Extender Terminology

Some terms used in this document are as follows:

- Fabric interface—A 10-Gigabit Ethernet uplink port that is designated for connection from the Fabric Extender to its parent switch. A fabric interface cannot be used for any other purpose. It must be directly connected to the parent switch.




---

**Note** A fabric interface includes the corresponding interface on the parent switch. This interface is enabled when you enter the **switchport mode fex-fabric** command.

---

- Port channel fabric interface—A port channel uplink connection from the Fabric Extender to its parent switch. This connection consists of fabric interfaces that are bundled into a single logical channel.
- Host interface—An Ethernet host interface for connection to a server or host system.




---

**Note** Do not connect a bridge or switch to a host interface. These interfaces are designed to provide end host or server connectivity.

---




---

**Note** On Cisco Nexus 2348TQ and Nexus 2348UPQ FEX, if a port channel is used to connect a parent switch with a Fabric Extender device, the port channels can have maximum of 8 ports.

The Nexus 2348 FEX devices have a total of 6 \* 40 Gigabit Ethernet uplink ports towards the parent switch. If these are used with native 40G uplinks port on a parent switch, then there is no limitation. All 6 ports can be used in either single homed or dual homed configuration. You can also use 40 Gigabit Ethernet uplink ports on the N2348 Fabric Extender device with 10 Gigabit Ethernet ports on the parent switch when used with the appropriate cabling. A maximum of 8 ports can be added to the port channel between the parent switch and Fabric Extender device. If it is a dual homed setup, VPC to the Fabric Extender device, only 4 ports per switch are allowed in the port channel.

---

- Port channel host interface—A port channel host interface for connection to a server or host system.

# Fabric Extender Features

The Cisco Nexus 2000 Series Fabric Extender allows a single switch—and a single consistent set of switch features—to be supported across a large number of hosts and servers. By supporting a large server-domain under a single management entity, policies can be enforced more efficiently.

Some of the features of the parent switch cannot be extended onto the Fabric Extender.

## Layer 2 Host Interfaces

The Fabric Extender provides connectivity for computer hosts and other edge devices in the network fabric.

Follow these guidelines when connecting devices to Fabric Extender host interfaces:

- All Fabric Extender host interfaces run as spanning tree edge ports with BPDU Guard enabled and you cannot configure them as spanning tree network ports.
- You can connect servers that use active/standby teaming, 802.3ad port channels, or other host-based link redundancy mechanisms to Fabric Extender host interfaces.
- Any device that is running spanning tree connected to a Fabric Extender host interface results in that host interface being placed in an error-disabled state when a BPDU is received.
- You can connect only virtual switches that leverages a link redundancy mechanism not dependent on spanning tree such as Cisco FlexLink or vPC (with the BPDU Filter enabled) to a Fabric Extender host interface. Because spanning tree is not used to eliminate loops, you should ensure a loop-free topology below the Fabric Extender host interfaces.

You can enable host interfaces to accept Cisco Discovery Protocol (CDP) packets. This protocol only works when it is enabled for both ends of a link.

**Note**

---

CDP is not supported on fabric interfaces when the Fabric Extender is configured in a virtual port channel (vPC) topology.

---

Ingress and egress packet counters are provided on each host interface.

For more information about BPDU Guard, see [Understanding BPDU Guard](#), on page 216 .

## Host Port Channel

The Cisco Nexus 2248TP, Cisco Nexus 2232PP, Cisco Nexus 2224TP, , Cisco Nexus B22 Fabric Extender for Fujitsu (N2K-B22FTS-P), Cisco Nexus B22 Fabric Extender for Dell (N2K-B22DELL-P), and Cisco Nexus B22 Fabric Extender for HP (N2K-B22HP-P) support port channel host interface configurations. Up to eight interfaces can be combined in a port channel. The port channel can be configured with or without LACP.

## VLANs and Private VLANs

The Fabric Extender supports Layer 2 VLAN trunks and IEEE 802.1Q VLAN encapsulation. Host interfaces can be members of private VLANs with the following restrictions:

- You can configure a host interface as an isolated or community access port only.
- You cannot configure a host interface as a promiscuous port.
- You cannot configure a host interface as a private VLAN trunk port.

For more information about VLANs, see the chapter in this guide on Configuring VLANs.

## Virtual Port Channels

With a virtual port channel (vPC), you can configure topologies where a Cisco Nexus Fabric Extender is connected to a pair of parent switches or a pair of Fabric Extenders are connected to a single parent switch. The vPC can provide multipath connections, which allow you to create redundancy between the nodes on your network.



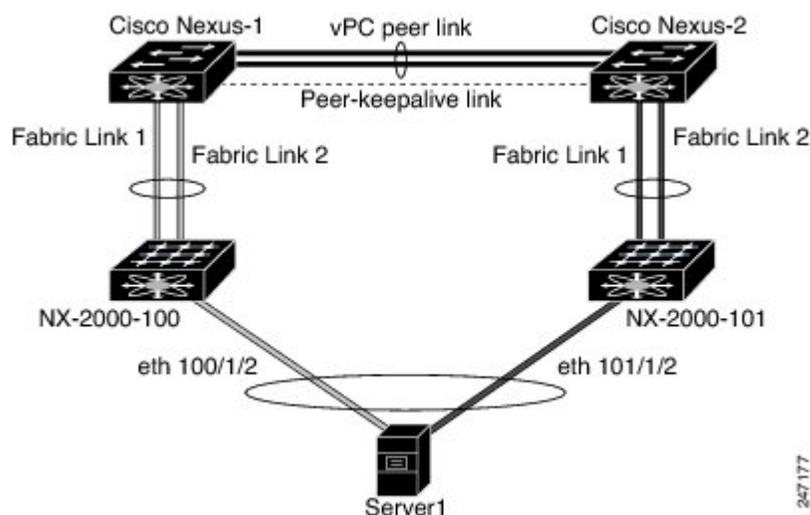
### Note

A port channel between two FEXs that are connected to the same Cisco Nexus device is not supported. Virtual port channels (vPCs) cannot span two different FEXs when connected to the same Cisco Nexus device.

The following vPC topologies are possible with the Fabric Extender:

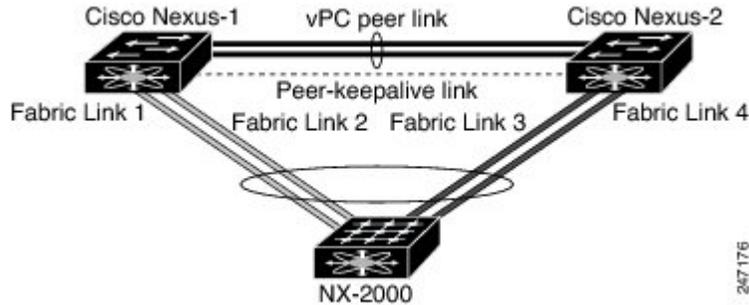
- The parent switches are connected single homed to Fabric Extenders that are subsequently connected to servers with dual interfaces (see the following figure).

**Figure 28: Single Homed Fabric Extender vPC Topology**



- The Fabric Extender is connected dual homed to two upstream parent switches and connected downstream to single homed servers (see the following figure).

**Figure 29: Dual Homed Fabric Extender vPC Topology**



This configuration is also called an Active-Active topology.



**Note**

Port channels between two Fabric Extenders connected to the same Cisco Nexus device is not supported vPCs cannot span two different Fabric Extenders that are connected to the same physical Cisco Nexus device.

## Fibre Channel over Ethernet Support

The Cisco Nexus 2232PP supports Fibre Channel over Ethernet (FCoE) with the following restrictions:

- Only FCoE Initialization Protocol (FIP) enabled converged network adapters (CNAs) are supported on the Fabric Extender.
- Binding to a port channel is limited to only one member in the port channel.

For configuration details, see the Fibre Channel over Ethernet Configuration Guide for the Nexus software release that you are using. The available versions of this document can be found at the following URL: [http://www.cisco.com/en/US/products/ps9670/products\\_installation\\_and\\_configuration\\_guides\\_list.html](http://www.cisco.com/en/US/products/ps9670/products_installation_and_configuration_guides_list.html).

## Protocol Offload

To reduce the load on the control plane of the Cisco Nexus Series device, Cisco NX-OS allows you to offload link-level protocol processing to the Fabric Extender CPU. The following protocols are supported:

- Link Layer Discovery Protocol (LLDP)
- Cisco Discovery Protocol (CDP)
- Link Aggregation Control Protocol (LACP)

## Quality of Service

### Access Control Lists

The Fabric Extender supports the full range of ingress access control lists (ACLs) that are available on its parent Cisco Nexus Series device.

### IGMP Snooping

### Switched Port Analyzer

### Fabric Interface Features

## Oversubscription

## Management Model

The Cisco Nexus 2000 Series Fabric Extender is managed by its parent switch over the fabric interfaces through a zero-touch configuration model. The switch discovers the Fabric Extender by detecting the fabric interfaces of the Fabric Extender.

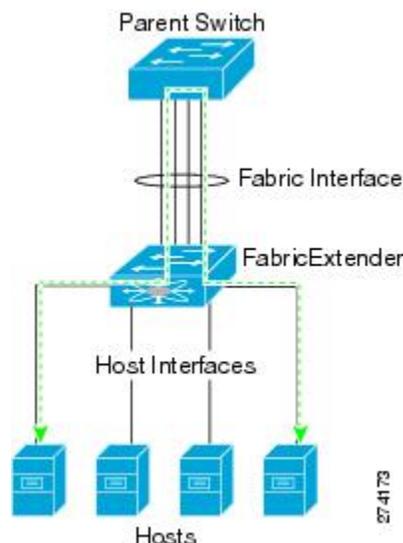
After discovery, if the Fabric Extender has been correctly associated with the parent switch, the following operations are performed:

- 1 The switch checks the software image compatibility and upgrades the Fabric Extender if necessary.
- 2 The switch and Fabric Extender establish in-band IP connectivity with each other.  
The switch assigns an IP address in the range of loopback addresses (127.15.1.0/24) to the Fabric Extender to avoid conflicts with IP addresses that might be in use on the network.
- 3 The switch pushes the configuration data to the Fabric Extender. The Fabric Extender does not store any configuration locally.
- 4 The Fabric Extender updates the switch with its operational status. All Fabric Extender information is displayed using the switch commands for monitoring and troubleshooting.

## Forwarding Model

The Cisco Nexus 2000 Series Fabric Extender does not perform any local switching. All traffic is sent to the parent switch that provides central forwarding and policy enforcement, including host-to-host communications between two systems that are connected to the same Fabric Extender as shown in the following figure.

**Figure 30: Forwarding Model**



The forwarding model facilitates feature consistency between the Fabric Extender and its parent Cisco Nexus Series device.



### Note

The Fabric Extender provides end-host connectivity into the network fabric. As a result, BPDU Guard is enabled on all its host interfaces. If you connect a bridge or switch to a host interface, that interface is placed in an error-disabled state when a BPDU is received.

You cannot disable BPDU Guard on the host interfaces of the Fabric Extender.

The Fabric Extender supports egress multicast replication from the network to the host. Packets that are sent from the parent switch for multicast addresses attached to the Fabric Extender are replicated by the Fabric Extender ASICs and are then sent to corresponding hosts.

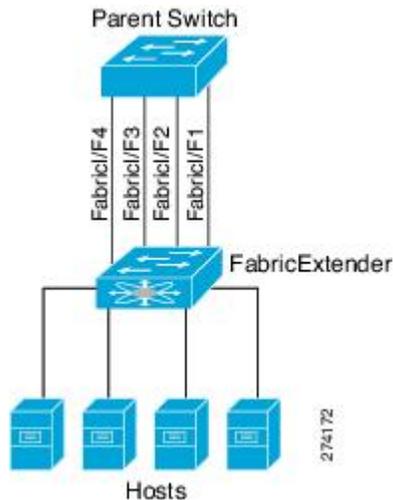
## Connection Model

Two methods (the static pinning fabric interface connection and the Port Channel fabric interface connection) allow the traffic from an end host to the parent switch to be distributed when going through the Cisco Nexus 2000 Series Fabric Extender.

## Static Pinning Fabric Interface Connection

To provide a deterministic relationship between the host interfaces and the parent switch, you can configure the Fabric Extender to use individual fabric interface connections. This configuration connects the 10-Gigabit Ethernet fabric interfaces as shown in the following figure. You can use any number of fabric interfaces up to the maximum available on the model of the Fabric Extender.

**Figure 31: Static Pinning Fabric Interface Connections**



When the Fabric Extender is brought up, its host interfaces are distributed equally among the available fabric interfaces. As a result, the bandwidth that is dedicated to each end host toward the parent switch is never changed by the switch but instead is always specified by you.



### Note

If a fabric interface fails, all its associated host interfaces are brought down and remain down until the fabric interface is restored.

You must use the **pinning max-links** command to create a number of pinned fabric interface connections so that the parent switch can determine a distribution of host interfaces. The host interfaces are divided by the number of the max-links and distributed accordingly. The default value is max-links 1.



### Caution

Changing the value of the **max-links** is disruptive; all the host interfaces on the Fabric Extender are brought down and back up as the parent switch reassigns its static pinning.

The pinning order of the host interfaces is initially determined by the order in which the fabric interfaces were configured. When the parent switch is restarted, the configured fabric interfaces are pinned to the host interfaces in an ascending order by the port number of the fabric interface.

To guarantee a deterministic and sticky association across a reboot, you can manually redistribute the pinning.

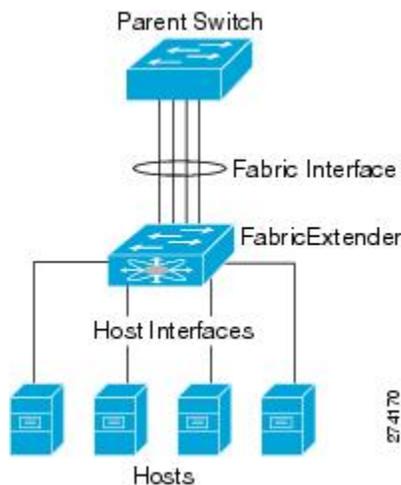
**Note**

The redistribution of the host interfaces will always be in an ascending order by the port number of the fabric interface.

## Port Channel Fabric Interface Connection

To provide load balancing between the host interfaces and the parent switch, you can configure the Fabric Extender to use a port channel fabric interface connection. This connection bundles 10-Gigabit Ethernet fabric interfaces into a single logical channel as shown in the following figure.

**Figure 32: Port Channel Fabric Interface Connection**



When you configure the Fabric Extender to use a port channel fabric interface connection to its parent switch, the switch load balances the traffic from the hosts that are connected to the host interface ports by using the following load-balancing criteria to select the link:

- For a Layer 2 frame, the switch uses the source and destination MAC addresses.
- For a Layer 3 frame, the switch uses the source and destination MAC addresses and the source and destination IP addresses.

**Note**

A fabric interface that fails in the port channel does not trigger a change to the host interfaces. Traffic is automatically redistributed across the remaining links in the port channel fabric interface. If all links in the fabric port channel go down, all host interfaces on the FEX are set to the down state.

# Port Numbering Convention

## Fabric Extender Image Management

No software ships with the Cisco Nexus 2000 Series Fabric Extender. The Fabric Extender image is bundled into the system image of the parent switch. The image is automatically verified and updated (if required) during the association process between the parent switch and the Fabric Extender.

When you enter the **install all** command, it upgrades the software on the parent Cisco Nexus Series switch and also upgrades the software on any attached Fabric Extender. To minimize downtime as much as possible, the Fabric Extender remains online while the installation process loads its new software image. Once the software image has successfully loaded, the parent switch and the Fabric Extender both automatically reboot.

This process is required to maintain version compatibility between the parent switch and the Fabric Extender.

## Fabric Extender Hardware

The Cisco Nexus 2000 Series Fabric Extender architecture allows hardware configurations with various host interface counts and speeds.

## Chassis

The Cisco Nexus 2000 Series Fabric Extender is a 1 RU chassis that is designed for rack mounting. The chassis supports redundant hot-swappable fans and power supplies.

## Ethernet Interfaces

There are four models of the Cisco Nexus 2000 Series Fabric Extender:

- The Cisco Nexus 2148T has 48 1000BASE-T Ethernet host interfaces for its downlink connection to servers or hosts and 4 10-Gigabit Ethernet fabric interfaces with SFP+ interface adapters for its uplink connection to the parent switch.
- The Cisco Nexus 2224TP has 24 100BASE-T/1000Base-T Ethernet host interfaces for its downlink connection to servers or hosts and 2 10-Gigabit Ethernet fabric interfaces with SFP+ interface adapters for its uplink connection to the parent switch.
- The Cisco Nexus 2232PP has 32 10-Gigabit Ethernet host interfaces with SFP+ interface adapters and 8 10-Gigabit Ethernet fabric interfaces with SFP+ interface adapters for its uplink connection to the parent switch.
- The Cisco Nexus 2248TP has 48 100BASE-T/1000Base-T Ethernet host interfaces for its downlink connection to servers or hosts and 4 10-Gigabit Ethernet fabric interfaces with SFP+ interface adapters for its uplink connection to the parent switch.

The Cisco Nexus 2248TP-E has all the features of the Cisco Nexus 2248TP with these additional features:

- A larger buffer to absorb large bursts.

- Support for an ingress and egress queue-limit per port.
  - Support for debug counters.
  - Support for pause no-drop behavior over a cable distance of 3000 meters between the Fabric Extender and switch.
  - Support for a user configurable shared-buffer.
- The Cisco Nexus B22 Fabric Extender for HP (NB22HP) has 16 1G/10-Gigabit Ethernet host interfaces. All host interfaces use all of the available fabric interfaces.
  - The Cisco Nexus B22 Fabric Extender for Fujitsu (NB22FTS) has 16 10-Gigabit Ethernet host interfaces. All host interfaces use all of the available fabric interfaces.
  - The Cisco Nexus B22 Fabric Extender for Dell (NB22DELL) has 16 1G/10-Gigabit Ethernet host interfaces. All host interfaces use all of the available fabric interfaces.

## Associating a Fabric Extender to a Fabric Interface

### Associating a Fabric Extender to an Ethernet Interface

#### Before You Begin

Ensure that you have enabled the Fabric Extender feature.

#### SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet *slot/port***
3. **switchport mode fex-fabric**
4. **fex associate *FEX-number***
5. (Optional) **show interface ethernet *port/slot* fex-intf**

#### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>configure terminal</b>  <b>Example:</b> <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	<b>interface ethernet <i>slot/port</i></b>  <b>Example:</b> <pre>switch(config)# interface ethernet 1/40 switch(config)#</pre>	Specifies an Ethernet interface to configure.

	Command or Action	Purpose
<b>Step 3</b>	<b>switchport mode fex-fabric</b>  <b>Example:</b> switch(config-if)# switchport mode fex-fabric switch(config-if)#	Sets the interface to support an external Fabric Extender.
<b>Step 4</b>	<b>fex associate FEX-number</b>  <b>Example:</b> switch(config-if)# fex associate 101 switch#	Associates the FEX number to the Fabric Extender unit attached to the interface. The range of the FEX number is from 100 to 199.
<b>Step 5</b>	<b>show interface ethernet port/slot fex-intf</b>  <b>Example:</b> switch# show interface ethernet 1/40 fex-intf switch#	(Optional) Displays the association of a Fabric Extender to an Ethernet interface.

This example shows how to associate the Fabric Extender to an Ethernet interface on the parent device:

```
switch# configure terminal
switch(config)# interface ethernet 1/40
switch(config-if)# switchport mode fex-fabric
switch(config-if)# fex associate 101
switch(config)#
```

This example shows how to display the association of the Fabric Extender and the parent device:

```
switch# show interface ethernet 1/40 fex-intf
Fabric          FEX
Interface      Interfaces
-----
Eth1/40        Eth101/1/48  Eth101/1/47  Eth101/1/46  Eth101/1/45
                Eth101/1/44  Eth101/1/43  Eth101/1/42  Eth101/1/41
                Eth101/1/40  Eth101/1/39  Eth101/1/38  Eth101/1/37
                Eth101/1/36  Eth101/1/35  Eth101/1/34  Eth101/1/33
                Eth101/1/32  Eth101/1/31  Eth101/1/30  Eth101/1/29
                Eth101/1/28  Eth101/1/27  Eth101/1/26  Eth101/1/25
                Eth101/1/24  Eth101/1/23  Eth101/1/22  Eth101/1/21
                Eth101/1/20  Eth101/1/19  Eth101/1/18  Eth101/1/17
                Eth101/1/16  Eth101/1/15  Eth101/1/14  Eth101/1/13
                Eth101/1/12  Eth101/1/11  Eth101/1/10  Eth101/1/9
                Eth101/1/8   Eth101/1/7   Eth101/1/6   Eth101/1/5
                Eth101/1/4   Eth101/1/3   Eth101/1/2   Eth101/1/1
```

## Associating a Fabric Extender to a Port Channel

### Before You Begin

Ensure that you have enabled the Fabric Extender feature.

## SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *channel*
3. **switchport mode fex-fabric**
4. **fex associate** *FEX-number*
5. (Optional) **show interface port-channel** *channel fex-intf*

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>configure terminal</b>  <b>Example:</b> switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	<b>interface port-channel</b> <i>channel</i>  <b>Example:</b> switch(config)# interface port-channel 4 switch(config-if)#	Specifies a port channel to configure.
Step 3	<b>switchport mode fex-fabric</b>  <b>Example:</b> switch(config-if)# switchport mode fex-fabric	Sets the port channel to support an external Fabric Extender.
Step 4	<b>fex associate</b> <i>FEX-number</i>  <b>Example:</b> switch(config-if)# fex associate 101	Associates a FEX number to the Fabric Extender unit attached to the interface. The range is from 101 to 199.
Step 5	<b>show interface port-channel</b> <i>channel fex-intf</i>  <b>Example:</b> switch# show interface port-channel 4 fex-intf	(Optional) Displays the association of a Fabric Extender to a port channel interface.

This example shows how to associate the Fabric Extender to a port channel interface on the parent device:

```
switch# configure terminal
switch(config)# interface ethernet 1/28
switch(config-if)# channel-group 4
switch(config-if)# no shutdown
switch(config-if)# exit
switch(config)# interface ethernet 1/29
switch(config-if)# channel-group 4
switch(config-if)# no shutdown
switch(config-if)# exit
switch(config)# interface ethernet 1/30
switch(config-if)# channel-group 4
switch(config-if)# no shutdown
switch(config-if)# exit
switch(config)# interface ethernet 1/31
```

```

switch(config-if)# channel-group 4
switch(config-if)# no shutdown
switch(config-if)# exit
switch(config)# interface port-channel 4
switch(config-if)# switchport
switch(config-if)# switchport mode fex-fabric
switch(config-if)# fex associate 101

```

**Tip**

As a best practice, only enter the **fex associate** command from the port channel interface, not from the physical interface.

**Note**

When adding physical interfaces to port channels, all configurations on the port channel and physical interface must match.

This example shows how to display the association of the Fabric Extender and the parent device:

```

switch# show interface port-channel 4 fex-intf
Fabric          FEX
Interface       Interfaces
-----
Po4             Eth101/1/48  Eth101/1/47  Eth101/1/46  Eth101/1/45
                Eth101/1/44  Eth101/1/43  Eth101/1/42  Eth101/1/41
                Eth101/1/40  Eth101/1/39  Eth101/1/38  Eth101/1/37
                Eth101/1/36  Eth101/1/35  Eth101/1/34  Eth101/1/33
                Eth101/1/32  Eth101/1/31  Eth101/1/30  Eth101/1/29
                Eth101/1/28  Eth101/1/27  Eth101/1/26  Eth101/1/25
                Eth101/1/24  Eth101/1/23  Eth101/1/22  Eth101/1/21
                Eth101/1/20  Eth101/1/19  Eth101/1/18  Eth101/1/17
                Eth101/1/16  Eth101/1/15  Eth101/1/14  Eth101/1/13
                Eth101/1/12  Eth101/1/11  Eth101/1/10  Eth101/1/9
                Eth101/1/8   Eth101/1/7   Eth101/1/6   Eth101/1/5
                Eth101/1/4   Eth101/1/3   Eth101/1/2   Eth101/1/1

```

## Disassociating a Fabric Extender from an Interface

### Before You Begin

Ensure that you have enabled the Fabric Extender feature.

### SUMMARY STEPS

1. **configure terminal**
2. **interface** {*ethernet slot/port* | *port-channel channel*}
3. **no fex associate**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>configure terminal</b>  <b>Example:</b> <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	<b>interface {ethernet slot/port   port-channel channel}</b>  <b>Example:</b> <pre>switch(config)# interface port-channel 4 switch(config-if)#</pre>	Specifies the interface to configure. The interface can be an Ethernet interface or a port channel.
Step 3	<b>no fex associate</b>  <b>Example:</b> <pre>switch(config-if)# no fex associate</pre>	Disassociates the Fabric Extender unit attached to the interface.

## Configuring Fabric Extender Global Features

You can configure global features on the Fabric Extender.

### Before You Begin

Ensure that you have enabled the Fabric Extender feature set.

## SUMMARY STEPS

1. **configure terminal**
2. **fex** *FEX-number*
3. (Optional) **description** *desc*
4. (Optional) **no description**
5. (Optional) **no type**
6. (Optional) **pinning max-links** *uplinks*
7. (Optional) **no pinning max-links**
8. (Optional) **serial** *serial*
9. (Optional) **no serial**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>configure terminal</b>  <b>Example:</b> <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	<b>fex <i>FEX-number</i></b>  <b>Example:</b> <pre>switch(config)# fex 101 switch(config-fex)#</pre>	Enters FEX configuration mode for the specified Fabric Extender. The range of the <i>FEX-number</i> is from 100 to 199.
Step 3	<b>description <i>desc</i></b>  <b>Example:</b> <pre>switch(config-fex)# description Rack7A-N2K</pre>	(Optional) Specifies the description. The default is the string FEXxxxx where xxxx is the FEX number. If the FEX number is 123, the description is FEX0123.
Step 4	<b>no description</b>  <b>Example:</b> <pre>switch(config-fex)# no description</pre>	(Optional) Deletes the description.
Step 5	<b>no type</b>  <b>Example:</b> <pre>switch(config-fex)# no type</pre>	(Optional) Deletes the FEX type. When a Fabric Extender is connected to the fabric interfaces and does not match the configured type that is saved in the binary configuration on the parent switch, all configurations for all interfaces on the Fabric Extender are deleted.
Step 6	<b>pinning max-links <i>uplinks</i></b>  <b>Example:</b> <pre>switch(config-fex)# pinning max-links 2</pre>	(Optional) Defines the number of uplinks. The default is 1. The range is from 1 to 4.  This command is only applicable if the Fabric Extender is connected to its parent switch using one or more statically pinned fabric interfaces. There can only be one port channel connection.  <b>Caution</b> Changing the number of uplinks with the <b>pinning max-links</b> command disrupts all the host interface ports of the Fabric Extender.
Step 7	<b>no pinning max-links</b>  <b>Example:</b> <pre>switch(config-fex)# no pinning max-links</pre>	(Optional) Resets the number of uplinks to the default.  <b>Caution</b> Changing the number of uplinks with the <b>no pinning max-links</b> command disrupts all the host interface ports of the Fabric Extender.
Step 8	<b>serial <i>serial</i></b>  <b>Example:</b> <pre>switch(config-fex)# serial JAF1339BDSK</pre>	(Optional) Defines a serial number string. If this command is configured, a switch allows the corresponding chassis ID to associate (using the <b>fex associate</b> command) only if the Fabric Extender reports a matching serial number string.

	Command or Action	Purpose
		<b>Caution</b> Configuring a serial number that does not match the specified Fabric Extender forces the Fabric Extender offline.
<b>Step 9</b>	<b>no serial</b>  <b>Example:</b> <code>switch(config-fex)# no serial</code>	(Optional) Deletes the serial number string.

## Enabling the Fabric Extender Locator LED

The locator beacon LED on the Fabric Extender allows you to locate a specific Fabric Extender in a rack.

### SUMMARY STEPS

1. **locator-led fex** *FEX-number*
2. (Optional) **no locator-led fex** *FEX-number*

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>locator-led fex</b> <i>FEX-number</i>  <b>Example:</b> <code>switch# locator-led fex 101</code>	Turns on the locator beacon LED for a specific Fabric Extender.
<b>Step 2</b>	<b>no locator-led fex</b> <i>FEX-number</i>  <b>Example:</b> <code>switch# no locator-led fex 101</code>	(Optional) Turns off the locator beacon LED for a specific Fabric Extender.

## Redistributing the Links

When you provision the Fabric Extender with statically pinned interfaces, the downlink host interfaces on the Fabric Extender are pinned to the fabric interfaces in the order they were initially configured. If you want to maintain a specific relationship of host interfaces to fabric interface across reboots, you should repin the links.

You may want to perform this function in these two situations:

- A change in the max-links configuration.
- If you need to maintain the pinning order of host interfaces to fabric interfaces.

## Changing the Number of Links

If you initially configured a specific port on the parent switch, for example port 33, as your only fabric interface, all 48 host interfaces are pinned to this port. If you provision another port, for example 35, then you must enter the **pinning max-links 2** command to redistribute the host interfaces. All host interfaces are brought down and host interfaces 1 to 24 are pinned to fabric interface 33 and host interfaces 25 to 48 are pinned to fabric interface 35.

## Maintaining the Pinning Order

The pinning order of the host interfaces is initially determined by the order in which the fabric interfaces were configured. In this example, four fabric interfaces were configured in the following order:

```
switch# show interface ethernet 1/35 fex-intf
Fabric          FEX
Interface       Interfaces
-----
Eth1/35         Eth101/1/12  Eth101/1/11  Eth101/1/10  Eth101/1/9
                Eth101/1/8   Eth101/1/7   Eth101/1/6   Eth101/1/5
                Eth101/1/4   Eth101/1/3   Eth101/1/2   Eth101/1/1

switch# show interface ethernet 1/33 fex-intf
Fabric          FEX
Interface       Interfaces
-----
Eth1/33         Eth101/1/24  Eth101/1/23  Eth101/1/22  Eth101/1/21
                Eth101/1/20  Eth101/1/19  Eth101/1/18  Eth101/1/17
                Eth101/1/16  Eth101/1/15  Eth101/1/14  Eth101/1/13

switch# show interface ethernet 1/38 fex-intf
Fabric          FEX
Interface       Interfaces
-----
Eth1/38         Eth101/1/36  Eth101/1/35  Eth101/1/34  Eth101/1/33
                Eth101/1/32  Eth101/1/31  Eth101/1/30  Eth101/1/29
                Eth101/1/28  Eth101/1/27  Eth101/1/26  Eth101/1/25

switch# show interface ethernet 1/40 fex-intf
Fabric          FEX
Interface       Interfaces
-----
Eth1/40         Eth101/1/48  Eth101/1/47  Eth101/1/46  Eth101/1/45
                Eth101/1/44  Eth101/1/43  Eth101/1/42  Eth101/1/41
                Eth101/1/40  Eth101/1/39  Eth101/1/38  Eth101/1/37
```

The next time that you reboot the Fabric Extender, the configured fabric interfaces are pinned to the host interfaces in an ascending order by port number of the fabric interface. If you want to configure the same fixed distribution of host interfaces without restarting the Fabric Extender, enter the **fex pinning redistribute** command.

## Redistributing Host Interfaces



### Caution

This command disrupts all the host interface ports of the Fabric Extender.

## SUMMARY STEPS

1. **configure terminal**
2. **fex pinning redistribute** *FEX-number*

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>configure terminal</b>  <b>Example:</b> switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	<b>fex pinning redistribute</b> <i>FEX-number</i>  <b>Example:</b> switch(config) # <b>fex pinning redistribute 101</b> switch(config) #	Redistributes the host connections. The range of <i>FEX-number</i> is from 100 to 199.

## Verifying the Fabric Extender Configuration

Use the following commands to display configuration information about the defined interfaces on a Fabric Extender:

Command or Action	Purpose
<b>show fex</b> [ <i>FEX-number</i> ] [ <b>detail</b> ]	Displays information about a specific Fabric Extender or all attached units.
<b>show interface</b> <i>type number fex-intf</i>	Displays the Fabric Extender ports that are pinned to a specific switch interface.
<b>show interface fex-fabric</b>	Displays the switch interfaces that have detected a Fabric Extender uplink.
<b>show interface ethernet</b> <i>number transceiver</i> [ <b>fex-fabric</b> ]	Displays the SFP+ transceiver and diagnostic optical monitoring (DOM) information for the Fabric Extender uplinks.
<b>show feature-set</b>	Displays the status of the feature sets on the device.

## Configuration Examples for the Fabric Extender

This example shows how to display all the attached Fabric Extender units:

```
switch# show fex
      FEX          FEX          FEX          FEX
Number  Description      State      Model      Serial
-----
100     FEX0100             Online     N2K-C2248TP-1GE  JAF1339BDSK
101     FEX0101             Online     N2K-C2232P-10GE  JAF1333ADDD
102     FEX0102             Online     N2K-C2232P-10GE  JAS12334ABC
```

This example shows how to display the detailed status of a specific Fabric Extender:

```
switch# show fex 100 detail
FEX: 100 Description: FEX0100 state: Online
FEX version: 5.0(2)N1(1) [Switch version: 5.0(2)N1(1)]
FEX Interim version: 5.0(2)N1(0.205)
Switch Interim version: 5.0(2)N1(0.205)
Extender Model: N2K-C2224TP-1GE, Extender Serial: JAF1427BQLG
Part No: 73-13373-01
Card Id: 132, Mac Addr: 68:ef:bd:62:2a:42, Num Macs: 64
Module Sw Gen: 21 [Switch Sw Gen: 21]
post level: complete
pinning-mode: static Max-links: 1
Fabric port for control traffic: Eth1/29
Fabric interface state:
  Po100 - Interface Up. State: Active
  Eth1/29 - Interface Up. State: Active
  Eth1/30 - Interface Up. State: Active
Fex Port      State Fabric Port Primary Fabric
-----
Eth100/1/1    Up    Po100      Po100
Eth100/1/2    Up    Po100      Po100
Eth100/1/3    Up    Po100      Po100
Eth100/1/4    Up    Po100      Po100
Eth100/1/5    Up    Po100      Po100
Eth100/1/6    Up    Po100      Po100
Eth100/1/7    Up    Po100      Po100
Eth100/1/8    Up    Po100      Po100
Eth100/1/9    Up    Po100      Po100
Eth100/1/10   Up    Po100      Po100
Eth100/1/11   Up    Po100      Po100
Eth100/1/12   Up    Po100      Po100
Eth100/1/13   Up    Po100      Po100
Eth100/1/14   Up    Po100      Po100
Eth100/1/15   Up    Po100      Po100
Eth100/1/16   Up    Po100      Po100
Eth100/1/17   Up    Po100      Po100
Eth100/1/18   Up    Po100      Po100
Eth100/1/19   Up    Po100      Po100
Eth100/1/20   Up    Po100      Po100
Eth100/1/21   Up    Po100      Po100
Eth100/1/22   Up    Po100      Po100
Eth100/1/23   Up    Po100      Po100
Eth100/1/24   Up    Po100      Po100
Eth100/1/25   Up    Po100      Po100
Eth100/1/26   Up    Po100      Po100
Eth100/1/27   Up    Po100      Po100
Eth100/1/28   Up    Po100      Po100
Eth100/1/29   Up    Po100      Po100
Eth100/1/30   Up    Po100      Po100
Eth100/1/31   Up    Po100      Po100
Eth100/1/32   Up    Po100      Po100
Eth100/1/33   Up    Po100      Po100
Eth100/1/34   Up    Po100      Po100
Eth100/1/35   Up    Po100      Po100
Eth100/1/36   Up    Po100      Po100
Eth100/1/37   Up    Po100      Po100
Eth100/1/38   Up    Po100      Po100
Eth100/1/39   Up    Po100      Po100
Eth100/1/40   Down  Po100      Po100
Eth100/1/41   Up    Po100      Po100
Eth100/1/42   Up    Po100      Po100
```

```

Eth100/1/43    Up        Po100      Po100
Eth100/1/44    Up        Po100      Po100
Eth100/1/45    Up        Po100      Po100
Eth100/1/46    Up        Po100      Po100
Eth100/1/47    Up        Po100      Po100
Eth100/1/48    Up        Po100      Po100
    
```

```

Logs:
02/05/2010 20:12:17.764153: Module register received
02/05/2010 20:12:17.765408: Registration response sent
02/05/2010 20:12:17.845853: Module Online Sequence
02/05/2010 20:12:23.447218: Module Online
    
```

This example shows how to display the Fabric Extender interfaces pinned to a specific switch interface:

```

switch# show interface port-channel 100 fex-intf
Fabric          FEX
Interface       Interfaces
-----
Po100           Eth100/1/48  Eth100/1/47  Eth100/1/46  Eth100/1/45
                Eth100/1/44  Eth100/1/43  Eth100/1/42  Eth100/1/41
                Eth100/1/40  Eth100/1/39  Eth100/1/38  Eth100/1/37
                Eth100/1/36  Eth100/1/35  Eth100/1/34  Eth100/1/33
                Eth100/1/32  Eth100/1/31  Eth100/1/30  Eth100/1/29
                Eth100/1/28  Eth100/1/27  Eth100/1/26  Eth100/1/25
                Eth100/1/24  Eth100/1/22  Eth100/1/20  Eth100/1/19
                Eth100/1/18  Eth100/1/17  Eth100/1/16  Eth100/1/15
                Eth100/1/14  Eth100/1/13  Eth100/1/12  Eth100/1/11
                Eth100/1/10  Eth100/1/9   Eth100/1/8   Eth100/1/7
                Eth100/1/6   Eth100/1/5   Eth100/1/4   Eth100/1/3
                Eth100/1/2   Eth100/1/1
    
```

This example shows how to display the switch interfaces that are connected to a Fabric Extender uplink:

```

switch# show interface fex-fabric
Fabric          Fabric          Fex          FEX
Fex  Port         Port State    Uplink      Model        Serial
-----
100  Eth1/29       Active       3           N2K-C2248TP-1GE  JAF1339BDSK
100  Eth1/30       Active       4           N2K-C2248TP-1GE  JAF1339BDSK
102  Eth1/33       Active       1           N2K-C2232P-10GE  JAS12334ABC
102  Eth1/34       Active       2           N2K-C2232P-10GE  JAS12334ABC
102  Eth1/35       Active       3           N2K-C2232P-10GE  JAS12334ABC
102  Eth1/36       Active       4           N2K-C2232P-10GE  JAS12334ABC
101  Eth1/37       Active       5           N2K-C2232P-10GE  JAF1333ADDD
101  Eth1/38       Active       6           N2K-C2232P-10GE  JAF1333ADDD
101  Eth1/39       Active       7           N2K-C2232P-10GE  JAF1333ADDD
101  Eth1/40       Active       8           N2K-C2232P-10GE  JAF1333ADDD
    
```

This example shows how to display the SFP+ transceiver and diagnostic optical monitoring (DOM) information for Fabric Extender uplinks for an SFP+ transceiver that is plugged into the parent switch interface:

```

switch# show interface ethernet 1/40 transceiver
Ethernet1/40
  sfp is present
  name is CISCO-MOLEX INC
  part number is 74752-9026
  revision is A0
  serial number is MOC13321057
  nominal bitrate is 12000 Mbits/sec
  Link length supported for copper is 3 m(s)
  cisco id is --
  cisco extended id number is 4
    
```

This example shows how to display the SFP+ transceiver and DOM information for Fabric Extender uplinks for an SFP+ transceiver that is plugged into the uplink port on the Fabric Extender:

```

switch# show interface ethernet 1/40 transceiver fex-fabric
Ethernet1/40
  sfp is present
  name is CISCO-MOLEX INC
  part number is 74752-9026
  revision is A0
    
```

```

serial number is MOC13321057
nominal bitrate is 12000 Mbits/sec
Link length supported for 50/125mm fiber is 0 m(s)
Link length supported for 62.5/125mm fiber is 0 m(s)
cisco id is --
cisco extended id number is 4

```

## Verifying the Chassis Management Information

Use the following to display configuration information used on the switch supervisor to manage the Fabric Extender.

Command or Action	Purpose
<b>show diagnostic result fex</b> <i>FEX-number</i>	Displays results from the diagnostic test for a Fabric Extender.
<b>show environment fex</b> {all   <i>FEX-number</i> } [temperature   power   fan]	Displays the environmental sensor status.
<b>show inventory fex</b> <i>FEX-number</i>	Displays inventory information for a Fabric Extender.
<b>show module fex</b> [ <i>FEX-number</i> ]	Displays module information about a Fabric Extender.
<b>show sprom fex</b> <i>FEX-number</i> {all   backplane   powersupply <i>ps-num</i> }   all	Displays the contents of the serial PROM (SPROM) on the Fabric Extender.

### Configuration Examples for Chassis Management

This example shows how to display the module information about all connected Fabric Extender units:

```

switch# show module fex
FEX Mod Ports Card Type Model Status.
-----
100 1 48 Fabric Extender 48x1GE + 4x10G Mod N2K-C2248TP-1GE present
101 1 32 Fabric Extender 32x10GE + 8x10G Mo N2K-C2232P-10GE present
102 1 32 Fabric Extender 32x10GE + 8x10G Mo N2K-C2232P-10GE present

FEX Mod Sw Hw World-Wide-Name(s) (WWN)
-----
100 1 4.2(1)N1(1) 0.103 --
101 1 4.2(1)N1(1) 1.0 --
102 1 4.2(1)N1(1) 1.0 --

FEX Mod MAC-Address(es) Serial-Num
-----
100 1 000d.ece3.2800 to 000d.ece3.282f JAF1339BDSK
101 1 000d.ecca.73c0 to 000d.ecca.73df JAF1333ADDD
102 1 000d.ecd6.bec0 to 000d.ecd6.bedf JAS12334ABC

```

This example shows how to display the module information about a specific Fabric Extender:

```

switch# show module fex 100
FEX Mod Ports Card Type Model Status.
-----
100 1 48 Fabric Extender 48x1GE + 4x10G Mod N2K-C2248TP-1GE present

FEX Mod Sw Hw World-Wide-Name(s) (WWN)
-----
100 1 4.2(1)N1(1) 0.103 --

```

```
FEX Mod  MAC-Address(es)                               Serial-Num
-----  -
100 1    000d.ece3.2800 to 000d.ece3.282f             JAF1339BDSK
```

This example shows how to display the inventory information about a specific Fabric Extender:

```
switch# show inventory fex 101
NAME: "FEX 101 CHASSIS", DESCR: "N2K-C2248TP-1GE CHASSIS"
PID: N2K-C2248TP-1GE , VID: V00 , SN: SSI13380FSM

NAME: "FEX 101 Module 1", DESCR: "Fabric Extender Module: 48x1GE, 4x10GE Supervisor"
PID: N2K-C2248TP-1GE , VID: V00 , SN: JAF1339BDSK

NAME: "FEX 101 Fan 1", DESCR: "Fabric Extender Fan module"
PID: N2K-C2248-FAN , VID: N/A , SN: N/A

NAME: "FEX 101 Power Supply 2", DESCR: "Fabric Extender AC power supply"
PID: NXK-PAC-400W , VID: 000, SN: LIT13370QD6
```

This example shows how to display diagnostic test results for a specific Fabric Extender:

```
switch# show diagnostic result fex 101
FEX-101: 48x1GE/Supervisor SerialNo : JAF1339BDSK
Overall Diagnostic Result for FEX-101 : OK

Test results: (. = Pass, F = Fail, U = Untested)
TestPlatform:
0)          SPROM: -----> .
1)  Inband interface: -----> .
2)          Fan: -----> .
3)  Power Supply: -----> .
4) Temperature Sensor: -----> .

TestForwardingPorts:
Eth  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Port -----
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .

Eth  25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
Port -----
.  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .

TestFabricPorts:
Fabric 1  2  3  4
Port -----
.  .  .  .
```

This example shows how to display the environment status for a specific Fabric Extender:

```
switch# show environment fex 101

Temperature Fex 101:
-----
Module  Sensor      MajorThresh  MinorThres  CurTemp  Status
(Celsius) (Celsius)
-----
1       Outlet-1    60           50          33       ok
1       Outlet-2    60           50          38       ok
1       Inlet-1     50           40          35       ok
1       Die-1      100          90          44       ok

Fan Fex: 101:
-----
Fan      Model              Hw      Status
-----
Chassis  N2K-C2148-FAN     --      ok
PS-1     --                 --      absent
PS-2     NXK-PAC-400W     --      ok
```

Power Supply Fex 101:

-----  
 Voltage: 12 Volts  
 -----

PS	Model	Power (Watts)	Power (Amp)	Status
1	--	--	--	--
2	NXK-PAC-400W	4.32	0.36	ok

Mod	Model	Power Requested (Watts)	Power Requested (Amp)	Power Allocated (Watts)	Power Allocated (Amp)	Status
1	N2K-C2248TP-1GE	0.00	0.00	0.00	0.00	powered-up

Power Usage Summary:

Power Supply redundancy mode: redundant

Total Power Capacity 4.32 W

Power reserved for Supervisor(s) 0.00 W

Power currently used by Modules 0.00 W

Total Power Available 4.32 W  
 -----

This example shows how to display the SPROM for a specific Fabric Extender:

```
switch# show sprom fex 101 all
DISPLAY FEX 101 SUP sprom contents
Common block:
Block Signature : 0xabab
Block Version   : 3
Block Length    : 160
Block Checksum  : 0x1a1e
EEPROM Size     : 65535
Block Count     : 3
FRU Major Type  : 0x6002
FRU Minor Type  : 0x0
OEM String      : Cisco Systems, Inc.
Product Number  : N2K-C2248TP-1GE
Serial Number   : JAF1339BDSK
Part Number     : 73-12748-01
Part Revision   : 11
Mfg Deviation   : 0
H/W Version     : 0.103
Mfg Bits        : 0
Engineer Use    : 0
snmpOID         : 9.12.3.1.9.78.3.0
Power Consump   : 1666
RMA Code        : 0-0-0-0
CLEI Code       : XXXXXXXXTBDV00
VID             : V00
Supervisor Module specific block:
Block Signature : 0x6002
Block Version   : 2
Block Length    : 103
Block Checksum  : 0x2686
Feature Bits    : 0x0
HW Changes Bits : 0x0
Card Index      : 11016
MAC Addresses   : 00-00-00-00-00-00
Number of MACs  : 0
Number of EPLD : 0
Port Type-Num   : 1-48;2-4
Sensor #1       : 60,50
Sensor #2       : 60,50
```



```

00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00
00 00
License software-module specific block:
Block Signature : 0x6006
Block Version   : 1
Block Length    : 16
Block Checksum  : 0x86f
lic usage bits:
ff ff ff ff ff ff ff ff

DISPLAY FEX 101 power-supply 2 srom contents:
Common block:
Block Signature : 0xabab
Block Version   : 3
Block Length    : 160
Block Checksum  : 0x1673
EEPROM Size     : 65535
Block Count     : 2
FRU Major Type  : 0xab01
FRU Minor Type  : 0x0
OEM String      : Cisco Systems Inc   NXK-PAC-400W
Product Number  : NXK-PAC-400W
Serial Number   : LIT13370QD6
Part Number     : 341
Part Revision   : -037
CLEI Code       : 5-01 01 000
VID             : 000
snmpOID         : 12336.12336.12336.12336.12336.12336.12374.12336
H/W Version     : 43777.2
Current         : 36
RMA Code        : 200-32-32-32
Power supply specific block:
Block Signature : 0x0
Block Version   : 0
Block Length    : 0
Block Checksum  : 0x0
Feature Bits    : 0x0
Current 110v    : 36
Current 220v    : 36
Stackmib OID    : 0

```

## Configuring the Cisco Nexus N2248TP-E Fabric Extender

The Cisco Nexus 2248TP-E Fabric Extender supports all of the CLI commands of the Cisco Nexus 2248TP Fabric Extender with additional commands to configure the following:

- Shared buffer (FEX global level)
- Queue-limit in ingress direction (FEX global level and interface level)
- Queue-limit in egress direction (FEX global level and interface level)
- No drop class over a distance of 3000 meters between the FEX and switch (FEX global level)

## Configuring the Shared Buffer

The following are guidelines for the configuration of the shared buffer:

- Configuring the shared buffer is done at the FEX global level.
- The total available buffer is 32 MB which is shared in both the ingress and egress directions.
- The default size of the shared buffer is 2539 2KB.

However, when configuring an Ethernet-based pause no-drop class, the shared buffer size changes to 10800 KB. This change is required to increase the dedicated buffer that supports the pause no-drop class. The pause no-drop class does not use buffer space from the shared-pool.



**Note** Performing these commands might result in traffic disruption on all ports.

### SUMMARY STEPS

1. **configure terminal**
2. **fex chassis\_id**
3. **hardware N2248TP-E shared-buffer-size buffer-size**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>configure terminal</b>  <b>Example:</b> switch# configure terminal switch(config)#	Enters global configuration mode.
<b>Step 2</b>	<b>fex chassis_id</b>  <b>Example:</b> switch(config)# fax 100 switch(config-fex)#	Enters configuration mode for the specified FEX. The range of the <i>chassis_id</i> value is 100 to 199.
<b>Step 3</b>	<b>hardware N2248TP-E shared-buffer-size buffer-size</b>  <b>Example:</b> switch(config-fex)# hardware N2248TP-E shared-buffer-size 25000	Specifies the shared buffer size (KB). The range of the <i>buffer-size</i> value is 10800 KB to 2539 KB. <b>Note</b> The <b>hardware N2248TP-E shared-buffer-size</b> command specifies the default shared buffer size of 25392 KB.

**Example:**

```
switch# configure terminal
switch(config)# fax 100
```

```
switch(config-fex)# hardware N2248TP-E shared-buffer-size 25000
switch(config-fex)#
```

## Configuring the Queue-Limit at the Global Level

The following are guidelines for the configuration of the queue-limit:

- The tx queue limit specifies the buffer size used for each queue in the egress (n2h) direction.
- The rx queue limit specifies the buffer size used for each port in the ingress (h2n) direction.
- You can adjust the ingress queue limit when the FEX uplink experiences temporary congestion.
- You can adjust the egress queue limit for improved burst absorption or in a situation where there is a many to one traffic pattern.
- When you disable the tx queue-limit, any output port is able to use the entire shared buffer.

### SUMMARY STEPS

1. **configure terminal**
2. **fex chassis\_id**
3. **hardware N2248TP-E queue-limit queue-limit tx|rx**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>configure terminal</b>  <b>Example:</b> <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
<b>Step 2</b>	<b>fex chassis_id</b>  <b>Example:</b> <pre>switch(config)# fex 100 switch(config)#</pre>	Enters configuration mode for the specified FEX.  The range of the <i>chassis_id</i> value is 100 to 199.
<b>Step 3</b>	<b>hardware N2248TP-E queue-limit queue-limit tx rx</b>  <b>Example:</b> <pre>switch(config-fex)# hardware N2248TP-E queue-limit 83000 tx</pre>	Controls the egress (tx) or ingress (rx) queue tail drop threshold level on a FEX. <ul style="list-style-type: none"> <li>• The default queue-limit for tx (egress) is 4 MB.  <b>Note</b> The <b>hardware N2248TP-E queue-limit</b> command specifies the default tx queue-limit.</li> <li>• The default queue-limit for rx (ingress) is 1 MB.  <b>Note</b> The <b>hardware N2248TP-E queue-limit rx</b> command specifies the default rx queue-limit.</li> </ul>

**Example:**

```
switch# configure terminal
switch(config)# fex 100
switch(config-fex)# hardware N2248TP-E queue-limit 83000 tx
switch(config-fex)#
```

## Configuring the Queue-Limit at the Port Level

You can overwrite the global level configuration by configuring the queue-limit at the port level.

You can also disable the queue-limit at the port level.

### SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet chassis\_id / slot/port**
3. **hardware N2248TP-E queue-limit queue-limit tx|rx**

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>configure terminal</b>  <b>Example:</b> switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	<b>interface ethernet chassis_id / slot/port</b>  <b>Example:</b> switch(config)# interface ethernet 100/1/1	Enters interface configuration mode.
Step 3	<b>hardware N2248TP-E queue-limit queue-limit tx rx</b>  <b>Example:</b> switch(config-if)# hardware N2248TP-E queue-limit 83000 tx	Controls the egress (tx) or ingress (rx) queue tail drop threshold level on a FEX. <ul style="list-style-type: none"> <li>• The default queue-limit for tx (egress) is 4 MB.</li> <li>• The default queue-limit for rx (ingress) is 1 MB.</li> </ul>

**Example:**

```
switch# configure terminal
switch(config)# interface ethernet 100/1/1
switch(config-if)# hardware N2248TP-E queue-limit 83000 tx
switch(config-if)#
```

## Configuring Uplink Distance

The Cisco Nexus N2248TP-E FEX supports a pause no-drop class up to a distance of 3000 meters between the FEX and the switch.

The default cable length between the FEX and the switch is 300 meters.



**Note** When the pause no-drop class is not configured, the uplink distance configuration has no effect.

### SUMMARY STEPS

1. **configure terminal**
2. **fex chassis\_id**
3. **hardware N2248TP-E uplink-pause-no-drop distance distance-value**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>configure terminal</b>  <b>Example:</b> switch# configure terminal switch(config)#	Enters global configuration mode.
<b>Step 2</b>	<b>fex chassis_id</b>  <b>Example:</b> switch(config)# fex 100 switch(config-fex)#	Enters configuration mode for the specified FEX.  The range of the <i>chassis_id</i> value is 100 to 199.
<b>Step 3</b>	<b>hardware N2248TP-E uplink-pause-no-drop distance distance-value</b>  <b>Example:</b> switch(config-fex)# hardware N2248TP-E uplink-pause-no-drop distance 3000	Specifies the no-drop distance between the FEX and the switch.  The maximum distance is 3000 meters.  <b>Note</b> The <b>hardware N2248TP-E uplink-pause-no-drop distance</b> command specifies the default 300 meter cable length.

#### Example:

```
switch# configure terminal
switch(config)# fex 100
switch(config-fex)# hardware N2248TP-E uplink-pause-no-drop distance 3000
switch(config-fex)#
```



## INDEX

10-Gigabit Ethernet interface [278](#)  
1000BASE-T Ethernet interface [278](#)  
100BASE-T Ethernet interface [278](#)  
802.1Q VLANs [66, 77](#)  
    configuring [77](#)  
    private VLANs [66](#)

### A

access VLANs [71](#)  
    understanding [71](#)  
ACL support [274](#)  
active-active vPC topology [272](#)  
adding ports [89](#)  
    port channels [89](#)  
aging time, configuring [244](#)  
    MAC table [244](#)  
assigning [30](#)  
    port profile to a range of interfaces [30](#)  
associating fabric extender [279](#)

### B

blocking state, STP [171](#)  
BPDU guard [216](#)  
BPDU Guard [271, 275](#)  
bridge ID [162](#)  
broadcast storms [263](#)

### C

CDP [271, 273](#)  
changing max-links [286](#)  
channel mode [92](#)  
    port channels [92](#)  
channel modes [86](#)  
    port channels [86](#)  
chassis [278](#)

chassis configuration mode [283](#)  
chassis ID [278](#)  
Cisco Discovery Protocol, See [CDP](#)  
Cisco Nexus 2148T [278](#)  
Cisco Nexus 2224PP [278](#)  
Cisco Nexus 2232PP [278](#)  
Cisco Nexus 2248TP [278](#)  
Cisco Nexus B22 Fabric Extender for Fujitsu (NB22FTS) [278](#)  
Cisco Nexus B22 Fabric Extender for HP (NB22HP) [278](#)  
CIST regional root [190](#)  
CIST root [191](#)  
class of service, See [CoS](#)  
clearing dynamic addresses [245](#)  
    MAC table [245](#)  
community ports [52](#)  
community VLANs [52, 53](#)  
configuration data [274](#)  
configuration examples [236](#)  
    Flex Link [236](#)  
configuring [22, 33, 46, 62, 63, 93, 95](#)  
    description parameter [33](#)  
    error-disabled recovery interval [22](#)  
    isolated trunk port [63](#)  
    LACP fast timer rate [93](#)  
    LACP port priority [95](#)  
    promiscuous trunk port [62](#)  
    VLANs [46](#)  
configuring LACP [91](#)  
CoS [274](#)

### D

Data Center Bridging Exchange, See [DCBX](#)  
DCBX [273](#)  
debounce timer [13](#)  
    parameters [13](#)  
debounce timer, configuring [32](#)  
    Ethernet interfaces [32](#)  
default settings [231](#)  
    Flex Link [231](#)  
description [283](#)

digital optical monitoring, See **DOM**  
 disabling **17, 19, 34, 117**  
   CDP **19**  
   ethernet interfaces **34**  
   link negotiation **17**  
   vPCs **117**  
**DOM** **274**  
 drop queue **274**  
 dual homed fabric extender vPC topology **272**

## E

edge port (PortFast) **271**  
 enabling **19, 20, 21**  
   CDP **19**  
   error-disabled detection **20**  
   error-disabled recovery **21**  
 enhanced vPC **150, 151, 152, 153, 154, 155, 156, 157**  
   about **150**  
   configuration overview **152**  
   example configuration **157**  
   failure response **151**  
   licensing **152**  
   scalability **151**  
   supported platforms **150**  
   supported topologies **150**  
   verifying common port channel members **155**  
   verifying configuration **153**  
   verifying interface consistency **156**  
   verifying port channel numbers **154**  
 EtherChannel host interface **129**  
   creating **129**  
 Ethernet fabric interface **270**  
 Ethernet interface **278**  
 Ethernet interfaces **32**  
   debounce timer, configuring **32**  
 extended range VLANs **40**

## F

fabric extender **103**  
   terminology **103**  
 fabric interface **270**  
 fabric interface Port Channel **277**  
 fail-over load balancing **277**  
**FEX** **103**  
   terminology **103**  
 FEX trunk port **64**  
   pvlan **64**  
 FEX-number **278**

Flex Link **231, 233, 236**  
   configuration examples **236**  
   default settings **231**  
   preemption, configuring **233**  
 Flex Links **229, 230, 232**  
   configuring **232**  
   guidelines and limitations **230**  
   information about **229**

## G

graceful convergence **96, 97**  
   LACP **96, 97**  
   port channels **96, 97**  
     LACP **96, 97**  
       graceful convergence **96, 97**  
 guidelines and limitations **115, 230**  
   Flex Links **230**  
   vPCs **115**

## H

hardware hashing **91**  
   multicast traffic **91**  
 host interface **270**  
 host interface autonegotiation **274**  
 host interface flow control defaults **274**  
 host interface link-level flow control **274**  
 host ports **52**  
   kinds of **52**

## I

ICMPv2 **248**  
 IEEE 802.1p **274**  
 IEEE 802.1w **187**  
 IEEE 802.3x **274**  
**IGMP** **250**  
   snooping parameters, configuring **250**  
**IGMP forwarding** **249**  
**IGMP snooping** **249, 256, 274**  
   interoperation with MVR **256**  
   queries **249**  
**IGMPv1** **248**  
**IGMPv3** **249**  
 image management **278**  
 information about **229**  
   Flex Links **229**  
 interface information, displaying **34**  
   layer 2 **34**

interface speed [10, 17](#)  
     configuring [17](#)  
 interfaces [7, 8](#)  
     chassis ID [7](#)  
     options [7](#)  
     UDLD [8](#)  
 isolated port [52](#)  
 isolated VLANs [52, 53](#)

## J

jumbo frame [274](#)

## L

LACP [79, 85, 86, 87, 91, 96, 97, 273](#)  
     configuring [91](#)  
     graceful convergence [96, 97](#)  
         disabling [96](#)  
         reenabling [97](#)  
     marker responders [87](#)  
     port channels [85](#)  
     system ID [86](#)  
 LACP fast timer rate [93](#)  
     configuring [93](#)  
 LACP port priority [95](#)  
     configuring [95](#)  
 LACP-enabled vs static [88](#)  
     port channels [88](#)  
 LAN interface [73](#)  
     Ethernet access port [73](#)  
 layer 2 [34](#)  
     interface information, displaying [34](#)  
 layer 2 interfaces [14](#)  
     unified ports [14](#)  
 Layer 2 switching [3](#)  
     Ethernet switching [3](#)  
 LED beacon [285](#)  
 licensing [152, 256](#)  
     enhanced vPC [152](#)  
     MVR [256](#)  
 Link Aggregation Control Protocol [79](#)  
     See also [LACP](#)  
 Link Failure [174, 192](#)  
     detecting unidirectional [174](#)  
 Link Layer Discovery Protocol, See [LLDP](#)  
 LLDP [273](#)  
 load balancing [90](#)  
     port channels [90](#)  
         configuring [90](#)  
 local switching [275](#)

locator LED [285](#)  
 loopback address assignment [274](#)  
 loopback address range [274](#)

## M

MAC address configuration [245](#)  
     verifying [245](#)  
 MAC addresses [243](#)  
     static, configuring [243](#)  
 MAC table [244, 245](#)  
     aging time, configuring [244](#)  
     clearing dynamic addresses [245](#)  
 manual redistribution [276](#)  
 max-links disruption [276](#)  
 maximum transmission unit, See [MTU](#)  
 MST [190, 199](#)  
     CIST regional root [190](#)  
     setting to default values [199](#)  
 MSTP [187, 188, 189, 190, 191, 192, 199](#)  
     boundary ports [192](#)  
         described [192](#)  
     CIST regional root [190](#)  
     CIST root [191](#)  
     CIST, described [189](#)  
     CST [189, 190](#)  
         defined [189](#)  
         operations between regions [190](#)  
     IEEE 802.1s [190](#)  
         terminology [190](#)  
     IST [189, 190](#)  
         operations within a region [189](#)  
     mapping VLANs to MST instance [199](#)  
     MST region [187, 188, 189, 191](#)  
         CIST [189](#)  
         described [187](#)  
         hop-count mechanism [191](#)  
         supported spanning-tree instances [188](#)  
 MTU [274](#)  
 multicast replication [275](#)  
 multicast storms [263](#)  
 multicast traffic [91](#)  
     hardware hashing [91](#)  
     port channels [91](#)  
 MVR [255, 256, 257, 259, 260](#)  
     configuring global parameters [257](#)  
     configuring interfaces [259](#)  
     default settings [257](#)  
     guidelines and limitations [256](#)  
     interoperation with IGMP snooping [256](#)  
     interoperation with vPC snooping [256](#)  
     licensing [256](#)

MVR (*continued*)

- overview [255](#)
- verifying the configuration [260](#)

**N**

- native 802.1Q VLANs [77](#)
  - configuring [77](#)
- new in this release [1](#)
- no-drop queue [274](#)

**O**

- oversubscription [274](#)
- oversubscription ratio [274](#)

**P**

- packet counter [271](#)
- parameters, about [13](#)
  - debounce timer [13](#)
- per class flow control [274](#)
- PFC [274](#)
- physical Ethernet settings [37](#)
- pinning max-links [283](#)
- port channel [98, 277](#)
  - verifying configuration [98](#)
- port channel configuration [80](#)
  - guidelines and limitations [80](#)
- port channel fabric interface [270, 274](#)
- port channel host interface [270, 271](#)
- port channeling [79](#)
- port channels [79, 81, 82, 85, 88, 89, 90, 91, 92, 130](#)
  - adding ports [89](#)
  - channel mode [92](#)
  - compatibility requirements [81](#)
  - creating [88](#)
  - hardware hashing [91](#)
  - LACP [85](#)
  - LACP-enabled vs static [88](#)
  - load balancing [82, 90](#)
    - port channels [82](#)
  - moving into a vPC [130](#)
  - STP [79](#)
- port numbering [278](#)
- port profiles [11, 12](#)
  - about [11](#)
  - guidelines and limitations [12](#)
    - port profiles [12](#)
- PortFast BPDU filtering [217](#)

- ports [47](#)
  - adding to VLANs [47](#)
- preemption, configuring [233](#)
  - Flex Link [233](#)
- primary VLANs [52](#)
- priority flow control, See [PFC](#)
- private VLAN [272](#)
- private VLANs [52, 53, 55, 56, 66](#)
  - 802.1Q VLANs [66](#)
  - community VLANs [52, 53](#)
  - end station access to [56](#)
  - isolated trunk [55](#)
  - isolated VLANs [52, 53](#)
  - ports [52](#)
    - community [52](#)
    - isolated [52](#)
    - promiscuous [52](#)
  - primary VLANs [52](#)
  - promiscuous trunk [55](#)
  - secondary VLANs [52](#)
- promiscuous ports [52](#)
- pvlan [64](#)
  - FEX trunk port [64](#)

**Q**

- QoS [274](#)
- QoS broadcast class [274](#)
- QoS egress policies [274](#)
- QoS multicast class [274](#)
- quality-of-service, See [QoS](#)
- queue-limit [296, 297](#)
  - global level [296](#)
  - port level [297](#)

**R**

- rapid PVST priority [182](#)
- Rapid PVST+ [176](#)
  - configuring [176](#)
- rapid PVST+ configurations [186](#)
  - verifying [186](#)
- Rapid Spanning Tree Protocol [187](#)
- redistributing host interfaces [286](#)
- reduced MAC address [162](#)
- reserved-range VLANs [40](#)
- restarting [34](#)
  - ethernet interfaces [34](#)
- root guard [219](#)
- RSTP [166, 169, 173, 187](#)
  - active topology [169](#)

RSTP (*continued*)

- BPDU [173](#)
  - processing [173](#)
- designated port, defined [169](#)
- designated switch, defined [169](#)
- proposal-agreement handshake process [166](#)
- rapid convergence [166](#)
  - point-to-point links [166](#)
  - root ports [166](#)
- root port, defined [169](#)

**S**

- secondary VLANs [52](#)
- serial number [283](#)
- SFP+ [278](#)
- SFP+ interface adapter [278](#)
- SFP+ transceiver [10](#)
- SFP+ validation [274](#)
- shared buffer [295](#)
  - configuring [295](#)
- show diagnostics [290](#)
- show environment [290](#)
- show fabric interface [286](#)
- show fex [287](#)
- show inventory [290](#)
- show modules [290](#)
- show SPROM [290](#)
- show transceiver status [287](#)
- single homed fabric extender vPC topology [272](#)
- Small form-factor pluggable (plus) transceiver [10](#)
- small form-factor pluggable transceiver [278](#)
- snooping parameters, configuring [250](#)
  - IGMP [250](#)
- SPAN restrictions [274](#)
- SPAN source ports [274](#)
- static MAC addresses, configuring [243](#)
- static pinning [276](#)
- STP [79, 166, 171, 172, 215, 216](#)
  - edge ports [166, 215](#)
  - network ports [216](#)
  - normal ports [216](#)
  - port channel [79](#)
  - port types [215](#)
  - PortFast [166, 215](#)
  - understanding [171, 172](#)
    - Blocking State [171](#)
    - disabled state [172](#)
    - forwarding state [171](#)
    - learning state [171](#)
- STP bridge ID [162](#)
- STP overview [161](#)

- STP root guard [219](#)
- suspending orphan ports, secondary switch [128](#)
  - vPC topology [128](#)
- switchport fex-fabric mode [274](#)
- switchport saved configuration [274](#)

**T**

- terminology [103](#)
  - fabric extender [103](#)
- topology [104, 105](#)
  - dual homed fabric extender vPC [105](#)
  - single homed fabric extender vPC [104](#)
- traffic storms [265](#)
  - control [265](#)
- type [283](#)

**U**

- UDLD [8, 10](#)
  - aggressive mode [10](#)
  - defined [8](#)
  - nonaggressive mode [10](#)
- UDLD modeA [15](#)
  - configuring [15](#)
- understanding [71](#)
  - access VLANs [71](#)
- unicast storms [263](#)
- Unidirectional Link Detection [8](#)
- unified ports [8, 14](#)
  - configuring [14](#)
- uplink distance [298](#)
  - configuring [298](#)

**V**

- verifying [49, 186](#)
  - rapid PVST+ configurations [186](#)
  - VLAN configurations [49](#)
- version compatibility [278](#)
- VLAN configurations [49](#)
  - verifying [49](#)
- VLAN reserved ranges [44](#)
  - changing [44](#)
- VLANs [40, 44, 46, 47, 66](#)
  - adding ports to [47](#)
  - changing [44](#)
  - configuring [46](#)
  - extended range [40](#)
  - private [66](#)

- VLANs *(continued)*
  - reserved range [40](#)
- vPC [114, 150, 256](#)
  - enhanced [150](#)
  - interoperation with MVR [256](#)
  - with ARP or ND [114](#)
- vpc peer links [111](#)
  - interfaces [111](#)
- vPC peer switch topology [134, 136](#)
  - hybrid [136](#)
    - configuring [136](#)
- vPC peer switch topology *(continued)*
  - pure [134](#)
    - configuring [134](#)
- vPC peer switches [115](#)
- vPC terminology [103](#)
- vPC topology [128, 272](#)
  - suspending orphan ports, secondary switch [128](#)
- vPCs [115, 130](#)
  - guidelines and limitations [115](#)
  - moving port channels into [130](#)