Cisco Nexus 3600 NX-OS Interfaces Configuration Guide, Release 9.2x

First Published: 2018-07-16

Last Modified: 2023-09-11

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

CONTENTS

PREFACE

Preface ix

Audience ix

Document Conventions ix

Related Documentation for Cisco Nexus 3600 Platform Switches x

Documentation Feedback x

Communications, Services, and Additional Information x

CHAPTER 1

New and Changed Information 1

New and Changed Information 1

CHAPTER 2

Overview 3

Licensing Requirements 3

Supported Platforms 3

CHAPTER 3

Configuring Layer 2 Interfaces 5

Licensing Requirements 5

Information About Ethernet Interfaces 5

Interface Command 5

Unidirectional Link Detection Parameter 6

Default UDLD Configuration 6

UDLD Aggressive and Nonaggressive Modes 7

Guidelines and Limitations for Layer 2 Interfaces 7

Interface Speed 8

40-Gigabit Ethernet Interface Speed 8

SVI Autostate 9

Cisco Discovery Protocol 9

Default CDP Configuration 9
Error-Disabled State 10
Default Interfaces 11
Debounce Timer Parameters 11
MTU Configuration 11
Counter Values 11
Downlink Delay 12
Default Physical Ethernet Settings 13
Configuring Ethernet Interfaces 13
Guidelines for Configuring Ethernet Interfaces 13
Configuring the UDLD Mode 13
Triggering the Link State Consistency Checker 15
Configuring the Interface Speed 15
Configuring Breakout on QSFP 40-Gigabit Ethernet Interfaces 17
Disabling Link Negotiation 18
Disabling SVI Autostate 20
Configuring a Default Interface 21
Configuring the CDP Characteristics 22
Enabling or Disabling CDP 23
Enabling the Error-Disabled Detection 23
Enabling the Error-Disabled Recovery 24
Configuring the Error-Disabled Recovery Interval 25
Disabling the Error-Disabled Recovery 26
Configuring the Debounce Timer 27
Configuring the Description Parameter 28
Disabling and Restarting Ethernet Interfaces 28
Configuring MAC addresses Limitation on a VLAN 29
Configuring Custom EtherType or Tag Protocol Identifier (TPID) 30
Configuring Downlink Delay 31
Displaying Interface Information 32
CHAPTER 4 Configuring Layer 3 Interfaces 35
Information About Layer 3 Interfaces 35
Routed Interfaces 35

Subinterfaces 36
VLAN Interfaces 37
Changing VRF Membership for an Interface 38
Notes About Changing VRF Membership for an Interface 38
Loopback Interfaces 39
IP Unnumbered 39
Tunnel Interfaces 39
Guidelines and Limitations for Layer 3 Interfaces 39
Default Settings for Layer 3 Interfaces 40
SVI Autostate Disable 40
Configuring Layer 3 Interfaces 40
Configuring a Routed Interface 40
Configuring a Subinterface 41
Configuring the Bandwidth on an Interface 42
Configuring a VLAN Interface 43
Enabling Layer 3 Retention During VRF Membership Change 44
Configuring a Loopback Interface 45
Configuring IP Unnumbered on an Ethernet Interface 45
Assigning an Interface to a VRF 46
Configuring an Interface MAC Address 47
Configuring a MAC-Embedded IPv6 Address 48
Configuring SVI Autostate Disable 51
Configuring a DHCP Client on an Interface 52
Verifying the Layer 3 Interfaces Configuration 52
Monitoring Layer 3 Interfaces 53
Configuration Examples for Layer 3 Interfaces 55
Related Documents for Layer 3 Interfaces 55
CHAPTER 5 Configuring Port Channels 57
Information About Port Channels 57
Understanding Port Channels 58
Compatibility Requirements 58
Load Balancing Using Port Channels 60
Guidelines and Limitations for ECMP 61

Symmetric Hashing 62

Understanding LACP 62

LACP Overview 62

LACP ID Parameters 63

Channel Modes 64

LACP Marker Responders 65

LACP-Enabled and Static Port Channel Differences 65

LACP Port Channel Minimum Links and MaxBundle 65

Guidelines and Limitations 66

Configuring Port Channels 66

Creating a Port Channel 66

Adding a Port to a Port Channel 67

Configuring Load Balancing Using Port Channels 68

Enabling LACP 69

Configuring the Channel Mode for a Port 70

Configuring LACP Port Channel MinLinks 71

Configuring the LACP Port-Channel MaxBundle 72

Configuring the LACP Fast Timer Rate 74

Configuring the LACP System Priority and System ID 75

Configuring the LACP Port Priority 75

Disabling LACP Graceful Convergence 76

Reenabling LACP Graceful Convergence 77

Verifying Port Channel Configuration 79

Triggering the Port Channel Membership Consistency Checker 79

Verifying the Load-Balancing Outgoing Port ID 80

Port Profiles 81

Configuring Port Profiles 82

Creating a Port Profile 82

Entering Port-Profile Configuration Mode and Modifying a Port Profile 84

Assigning a Port Profile to a Range of Interfaces 84

Enabling a Specific Port Profile 85

Inheriting a Port Profile 86

Removing a Port Profile from a Range of Interfaces 87

Removing an Inherited Port Profile 88

CHAPTER 6

Configuring Virtual Port Channels 91
Information About vPCs 92
vPC Overview 92
Terminology 93
vPC Terminology 93
vPC Domain 93
Peer-Keepalive Link and Messages 94
Compatibility Parameters for vPC Peer Links 95
Configuration Parameters That Must Be Identical 95
Configuration Parameters That Should Be Identical 96
Per-VLAN Consistency Check 97
vPC Auto-Recovery 97
vPC Peer Links 97
vPC Peer Link Overview 98
vPC Number 99
vPC Interactions with Other Features 99
vPC and LACP 99
vPC Peer Links and STP 99
CFSoE 100
vPC Forklift Upgrade Scenario 100
Guidelines and Limitations for vPCs 103
Verifying the vPC Configuration 104
Viewing the Graceful Type-1 Check Status 104
Viewing a Global Type-1 Inconsistency 105
Viewing an Interface-Specific Type-1 Inconsistency 106
Viewing a Per-VLAN Consistency Status 107
vPC Default Settings 110
Configuring vPCs 110
Enabling vPCs 110
Disabling vPCs 111
Creating a vPC Domain 111
Configuring a vPC Keepalive Link and Messages 112
Creating a vPC Peer Link 115

Checking the Configuration Compatibility 116
Enabling vPC Auto-Recovery 117
Configuring the Restore Time Delay 118
Excluding VLAN Interfaces from Shutting Down a vPC Peer Link Fails 119
Configuring the VRF Name 119
Moving Other Port Channels into a vPC 120
Manually Configuring a vPC Domain MAC Address 121
Manually Configuring the System Priority 122
Manually Configuring a vPC Peer Switch Role 123
Configuring Layer 3 over vPC 124

Preface

This preface includes the following sections:

- Audience, on page ix
- Document Conventions, on page ix
- Related Documentation for Cisco Nexus 3600 Platform Switches, on page x
- Documentation Feedback, on page x
- Communications, Services, and Additional Information, on page x

Audience

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

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.

Convention	Description
`variable`	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Examples use the following conventions:

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

Related Documentation for Cisco Nexus 3600 Platform Switches

The entire Cisco Nexus 3600 platform switch documentation set is available at the following URL:

http://www.cisco.com/c/en/us/support/switches/nexus-3000-series-switches/tsd-products-support-series-home.html

Documentation Feedback

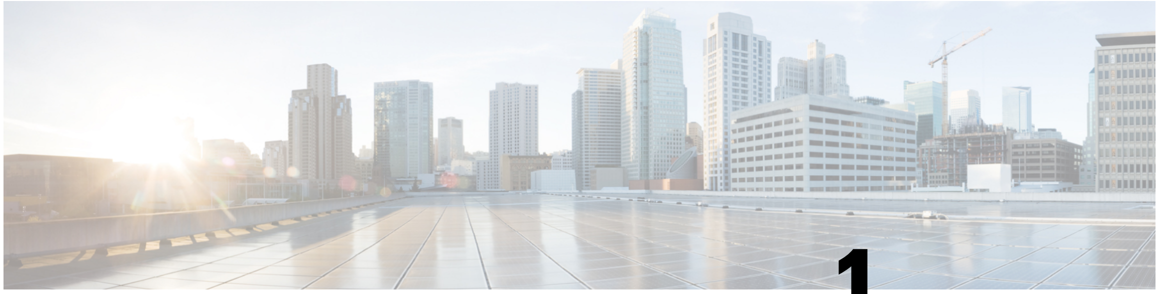
To provide technical feedback on this document, or to report an error or omission, please send your comments to nexus3k-docfeedback@cisco.com. We appreciate your feedback.

Communications, Services, and Additional Information

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

Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.



CHAPTER 1

New and Changed Information

- New and Changed Information, on page 1

New and Changed Information

Table 1: New and Changed Features

Feature	Description	Changed in Release	Where Documented
First 9x Release	First 9x Release	9.2(1)	N/A

CHAPTER 2

Overview

- Licensing Requirements, on page 3
- Supported Platforms, on page 3

Licensing Requirements

For a complete explanation of Cisco NX-OS licensing recommendations and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide* and the *Cisco NX-OS Licensing Options Guide*.

Supported Platforms

Starting with Cisco NX-OS release 7.0(3)I7(1), use the Nexus Switch Platform Support Matrix to know from which Cisco NX-OS releases various Cisco Nexus 9000 and 3000 switches support a selected feature.

CHAPTER 3

Configuring Layer 2 Interfaces

- Licensing Requirements, on page 5
- Information About Ethernet Interfaces, on page 5
- Guidelines and Limitations for Layer 2 Interfaces, on page 7
- Interface Speed, on page 8
- 40-Gigabit Ethernet Interface Speed, on page 8
- SVI Autostate, on page 9
- Cisco Discovery Protocol, on page 9
- Error-Disabled State, on page 10
- Default Interfaces, on page 11
- Debounce Timer Parameters, on page 11
- MTU Configuration, on page 11
- Default Physical Ethernet Settings , on page 13
- Configuring Ethernet Interfaces, on page 13
- Displaying Interface Information, on page 32

Licensing Requirements

For a complete explanation of Cisco NX-OS licensing recommendations and how to obtain and apply licenses, see the *Cisco NX-OS Licensing Guide* and the *Cisco NX-OS Licensing Options Guide*.

Information About Ethernet Interfaces

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

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:

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.

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.

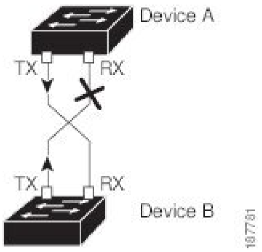
UDLD is a Layer 2 protocol that works with the Layer 1 protocols to determine the physical status of a link. At Layer 1, autonegotiation takes care of physical signaling and fault detection. UDLD performs tasks that autonegotiation cannot perform, such as detecting the identities of neighbors and shutting 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	Disabled on all Ethernet 10/100 and 1000BASE-TX LAN ports

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.

Guidelines and Limitations for Layer 2 Interfaces

Layer 2 interfaces have the following configuration guidelines and limitations:

- Auto-negotiation is not supported.
- 1G autonegotiation not supported on N3K-C36180YC-R and N9K-X96136YC-R switches. To work around this issue, you must manually set speed to 1000. If autonegotiation is enabled on the neighbors, you must disable autonegotiation on those neighbors.
- On Cisco Nexus N3K-C3636C-R and N3K-C36180YC-R switches, auto-negotiation may not work on ports 49-64 when bringing up 100G links using QSFP-100G-CR4 cable. To work around this issue, you must hard-code the speed on ports 49-64 and disable auto-negotiation

Interface Speed

Cisco Nexus 36180YC-R switches have 48 small form-factor pluggable (SFP) ports with a default speed of 10 G and 6 quad small form-factor pluggable (QSFP) ports with a default speed of 100 G. 48 SFP interface ports can support 25 G, 10 G, 1 G speeds. 6 QSFP interface ports can support 100 G and 40 G speeds.

In the first 48 ports, each 4 ports in the port group must have the same speed configured. You cannot configure one port at a time which might result in an error. For more information, see CSCve80686.

Table 3: Breakout Modes Support Matrix

Switches	4x10G	4x25G	2x50G
N3K-C3636C-R	Yes	Yes	Yes
N3K-C36180YC-R	Yes	Yes	Yes

40-Gigabit Ethernet Interface Speed

You can operate QSFP ports as either 40-Gigabit Ethernet or 4x10-Gigabit Ethernet modes on Cisco Nexus 3600 platform switches. By default, there are 6 QSFP interface ports numbered 49 to 54 which can operated in 40-Gigabit Ethernet mode. These 40-Gigabit Ethernet ports are numbered in a 2-tuple naming convention. For example, the second 40-Gigabit Ethernet port is numbered as 1/50. The process of changing the configuration from 40-Gigabit Ethernet to 10-Gigabit Ethernet is called breakout and the process of changing the configuration from 10-Gigabit Ethernet to Gigabit Ethernet is called breakin. When you break out a 40-Gigabit Ethernet port into 10-Gigabit Ethernet ports, the resulting ports are numbered using a 3-tuple naming convention. For example, the break-out ports of the second 40-Gigabit Ethernet port are numbered as 1/49/1, 1/49/2, 1/49/3, 1/49/4.

Note The breakout ports are in administratively enabled state after the breakout of the 40G ports into 4x10G mode or the breaking of the 100G ports into 4x25G mode. On upgrade from the earlier releases, the configuration restored takes care of restoring the appropriate administrative state of the ports.

Note When you break out from 40-Gigabit Ethernet to 10-Gigabit Ethernet, or break in from 10-Gigabit Ethernet to 40-Gigabit Ethernet, all interface configurations are reset, and the affected ports are administratively unavailable. To make these ports available, use the **no shut** command.

Note A new QSFP+ 40-Gb transceiver is supported on the Cisco Nexus 3600 platform switches. The new QSFP+ (40-Gb) transceiver has a cable that splits into four 10Gb SFP-10G-LR transceivers. To use it, you need the port to be in 4x10G mode. If you are using the breakout cable, you need to run that 40G port in 4x10G mode.

The ability to break out a 40-Gigabit Ethernet port into four 10-Gigabit Ethernet ports and break in four 10-Gigabit Ethernet ports into a 40-Gigabit Ethernet port dynamically allows you to use any of the breakout-capable ports to work in the 40-Gigabit Ethernet or 10-Gigabit Ethernet modes without permanently defining them.

SVI Autostate

The Switch Virtual Interface (SVI) represents a logical interface between the bridging function and the routing function of a VLAN in the device. By default, when a VLAN interface has multiple ports in the VLAN, the SVI goes to the down state when all the ports in the VLAN go down.

Autostate behavior is the operational state of an interface that is governed by the state of the various ports in its corresponding VLAN. An SVI interface on a VLAN comes up when there is at least one port in that vlan that is in STP forwarding state. Similarly, this interface goes down when the last STP forwarding port goes down or goes to another STP state.

By default, Autostate calculation is enabled. You can disable Autostate calculation for an SVI interface and change the default value.

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 4: Default CDP Configuration

Feature	Default Setting
CDP interface state	Enabled
CDP timer (packet update frequency)	60 seconds

Feature	Default Setting
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 inteface 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 reenable 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.

To disable recovery of an interface from the err-disabled state, use the **no errdisable recovery cause** command.

The various options for the **errdisable recover cause** command are as follows:

- all—Enables a timer to recover from all causes.
- bpduguard—Enables a timer to recover from the bridge protocol data unit (BPDU) Guard error-disabled state.
- failed-port-state—Enables a timer to recover from a Spanning Tree Protocol (STP) set port state failure.
- link-flap—Enables a timer to recover from linkstate flapping.
- pause-rate-limit—Enables a timer to recover from the pause rate limit error-disabled state.
- udld—Enables a timer to recover from the Unidirectional Link Detection (UDLD) error-disabled state.
- loopback—Enables a timer to recover from the loopback error-disabled state.

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.

Default Interfaces

You can use the default interface feature to clear the configured parameters for both physical and logical interfaces such as the Ethernet, loopback, management, VLAN, and the port-channel interface.

Debounce Timer Parameters

The debounce timer delays notification of a link change, which can decrease traffic loss due to network reconfiguration. You can configure the debounce timer separately for each Ethernet port and specify the delay time in milliseconds. The delay time can range from 0 milliseconds to 5000 milliseconds. By default, this parameter is set for 100 milliseconds, which results in the debounce timer not running. When this parameter is set to 0 milliseconds, the debounce timer is disabled.

Caution Enabling the debounce timer causes the link-down detections to be delayed, which results in a loss of traffic during the debounce period. This situation might affect the convergence and reconvergence of some Layer 2 and Layer 3 protocols.

MTU Configuration

The 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.

Counter Values

See the following information on the configuration, packet size, incremented counter values, and traffic.

Configuration	Packet Size	Incremented Counters	Traffic
L2 port – without any MTU configuration	6400 and 10000	Jumbo, giant, and input error	Dropped
L2 port – with jumbo MTU 9216 in network-qos configuration	6400	Jumbo	Forwarded
L2 port – with jumbo MTU 9216 in network-qos configuration	10000	Jumbo, giant, and input error	Dropped

Configuration	Packet Size	Incremented Counters	Traffic
Layer 3 port with default Layer 3 MTU and jumbo MTU 9216 in network-qos configuration	6400	Jumbo	Packets are punted to the CPU (subjected to CoPP configs), get fragmented, and then they are forwarded by the software.
Layer 3 port with default Layer 3 MTU and jumbo MTU 9216 in network-qos configuration	6400	Jumbo	Packets are punted to the CPU (subjected to CoPP configs), get fragmented, and then they are forwarded by the software.
Layer 3 port with default Layer 3 MTU and jumbo MTU 9216 in network-qos configuration	10000	Jumbo, giant, and input error	Dropped
Layer 3 port with jumbo Layer 3 MTU and jumbo MTU 9216 in network-qos configuration	6400	Jumbo	Forwarded without any fragmentation.
Layer 3 port with jumbo Layer 3 MTU and jumbo MTU 9216 in network-qos configuration	10000	Jumbo, giant, and input error	Dropped
Layer 3 port with jumbo Layer 3 MTU and default L2 MTU configuration	6400 and 10000	Jumbo, giant, and input error	Dropped

Note

- Under 64 bytes packet with good CRC–The short frame counter increments.
- Under 64 bytes packet with bad CRC–The runts counter increments.
- Greater than 64 bytes packet with bad CRC–The CRC counter increments.

Downlink Delay

You can operationally enable uplink SFP+ ports before downlink RJ-45 ports after a reload on a Cisco Nexus 3048 switch. You must delay enabling the RJ-45 ports in the hardware until the SFP+ ports are enabled.

You can configure a timer that during reload enables the downlink RJ-45 ports in hardware only after the specified timeout. This process allows the uplink SFP+ ports to be operational first. The timer is enabled in the hardware for only those ports that are admin-enable.

Downlink delay is disabled by default and must be explicitly enabled. When enabled, if the delay timer is not specified, it is set for a default delay of 20 seconds.

Default Physical Ethernet Settings

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

Parameter	Default Setting
Duplex	Auto (full-duplex)
Encapsulation	ARPA
MTU[1]	1500 bytes
Port Mode	Access
Speed	Auto (10000)

[1] MTU cannot be changed per-physical Ethernet interface. You modify MTU by selecting maps of QoS classes.

Configuring Ethernet Interfaces

Guidelines for Configuring Ethernet Interfaces

There is a behavior change in configuring the interface Ethernet commands on Cisco Nexus 3000 Series switches. For example, the command **sh int ethernet Eth1/1 transceiver** does not work anymore. You have to configure the command as **sh int ethernet 1/1 transceiver**.

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

Example

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
```

Triggering the Link State Consistency Checker

You can manually trigger the link state consistency checker to compare the hardware and software link status of an interface and display the results. To manually trigger the link state consistency checker and display the results, use the following command in any mode:

SUMMARY STEPS

1. switch# **show consistency-checker link-state module** *slot*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **show consistency-checker link-state module** *slot*	Starts a link state consistency check on the specified module and displays its results.

Example

This example shows how to trigger a Link State consistency check and display its results:

```
switch# show consistency-checker link-state module 1
Link State Checks: Link state only
Consistency Check: FAILED
No inconsistencies found for:
   Ethernet1/1
   Ethernet1/2
   Ethernet1/3
   Ethernet1/4
   Ethernet1/5
   Ethernet1/6
   Ethernet1/7
   Ethernet1/8
   Ethernet1/9
   Ethernet1/10
   Ethernet1/12
   Ethernet1/13
   Ethernet1/14
   Ethernet1/15
Inconsistencies found for following interfaces:
   Ethernet1/11
```

Configuring the Interface Speed

The first 48 ports support 1 G/10 G/25 G and the remaining 6 ports support 40 G/100 G.

In the first 48 ports, each 4 ports in the port group must have the same speed configured. You cannot configure one port at a time which might result in an error. For more information, see CSCve80686.

Table 5:

Port Groups	Ports
Port-Group 1	Ports 1-4
Port-Group 2	Ports 5-8
Port-Group 3	Ports 9-12
Port-Group 4	Ports 13-16
Port-Group 5	Ports 17-20
Port-Group 6	Ports 21-24
Port-Group 7	Ports 25-28
Port-Group 8	Ports 29-32
Port-Group 9	Ports 33-36
Port-Group 10	Ports 37-40
Port-Group 11	Ports 41-44
Port-Group 12	Ports 45-48

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 Gbps.

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
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Enters interface configuration mode for the specified interface. This interface must have a 1-Gigabit Ethernet SFP transceiver inserted into it.

	Command or Action	Purpose
Step 3	switch(config-if)# **speed** *speed*	Sets the speed on the interface. This command can only be applied to a physical Ethernet interface. The *speed* argument can be set to one of the following: • 10 Mbps • 100 Mbps • 1 Gbps • 10 Gbps • automatic

Example

This 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
```

Configuring Breakout on QSFP 40-Gigabit Ethernet Interfaces

When you break out ports into the 10-GbE mode, you can switch between the first QSFP port and SFP+ ports 1 to 4. Either the first QSFP port or the four SFP+ ports can be active at any time. QSFP is the default port with an interface speed of 40 Gbps.

When the first QSFP port is in the 40-GbE mode, you cannot switch the port to four SFP+ ports and the first QSFP port will be active until you break out the port into the 10-GbE mode. This is because SFP+ ports do not support the 40-GbE mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface breakout module** *module number* **port** *port range***map 10g-4x**
3. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface breakout module** *module number* **port** *port range***map 10g-4x**	Enables you to configure the module in 10g mode. When you are changing the portmode from QSFP to SFP+, the **hardware profile front portmode** command takes effect only after breaking out the first QSFP port as displayed in this command.

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

Example

This example shows how to configure breakout on 40-Gigabit Ethernet Interface:

```
switch# show int e1/49 tranceiver
Ethernet1/49transceiver is present
type is QSFP-4X10G-AOC1M
name is CISCO-AVAGO
part number is AFBR-7IER01Z-CS2
revision is 01
serial number is AVE20421070
nominal bitrate is 10300 MBit/sec per channel
Link length supported for copper is 1 m
cisco id is 13
cisco extended id number is 16
cisco part number is 10-2932-02
cisco product id is QSFP-4X10G-AOC1M
cisco vendor id is V02

switch# configure terminal
switch(config)#
switch(config)# interface breakout module 1 port 49 map 10g-4x
switch(config)# exit


switch# show interface ethernet 1/49/1-4 br

--------------------------------------------------------------------------------
Ethernet VLAN Type Mode Status Reason Speed Port
Interface Ch #
--------------------------------------------------------------------------------
Eth1/49/1 1 eth access up none 10G(D) --
Eth1/49/2 1 eth access up none 10G(D) --
Eth1/49/3 1 eth access up none 10G(D) --
Eth1/49/4 1 eth access up none 10G(D) --
```

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. By default, auto-negotiation is enabled on the Cisco Nexus 3064 and 3064-X switches and disabled on the Cisco Nexus 3048 switch. You cannot disable auto-negotiation on 1-Gigabit ports.

By default, auto-negotiation is enabled on all 1G SFP+ and 40G QSFP ports and it is disabled on 10G SFP+ ports. Auto-negotiation is by default enabled on all 1G and 10G Base-T ports. It cannot be disabled on 1G and 10G Base-T ports.

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

Starting with Release 6.0(2)U5(1), you can disable auto-negotiation on all 40G interfaces. A new CLI command **no system default interface 40g auto-negotation** is introduced to disable auto-negotiation across all the 40G interfaces. The new CLI command is only effective on the 40G interfaces and it does not have any effect on 1G or10G interfaces. For CR4 cables, the auto-negotiation configuration has to be identical at both the end devices for the link to come up.

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

Example

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)#
```

Disabling SVI Autostate

You can configure a SVI to remain active even if no interfaces are up in the corresponding VLAN. This enhancement is called Autostate Disable.

When you enable or disable autostate behavior, it is applied to all the SVIs in the switch unless you configure autostate per SVI .

Note Autostate behavior is enabled by default.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **feature interface-vlan**
3. switch(config)# **[no]system default interface-vlan autostate**
4. (Optional) switch(config)# **interface vlan** *interface-vlan-number*
5. (Optional) switch(config-if)# [**no**] **autostate**
6. (Optional) switch(config)# **show interface-vlan** *interface-vlan*
7. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature interface-vlan**	Enables the interface-vlan feature.
Step 3	Required: switch(config)# **[no]system default interface-vlan autostate**	Configures the system to enable or disable the Autostate default behavior.
Step 4	(Optional) switch(config)# **interface vlan** *interface-vlan-number*	Creates a VLAN interface. The number range is from 1 to 4094.
Step 5	(Optional) switch(config-if)# [**no**] **autostate**	Enables or disables Autostate behavior per SVI.
Step 6	(Optional) switch(config)# **show interface-vlan** *interface-vlan*	Displays the enabled or disabled Autostate behavior of the SVI.
Step 7	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to disable the systems Autostate default for all the SVIs on the switch:

```
switch# configure terminal
switch(config)# feature interface-vlan
switch(config)# system default interface-vlan no autostate
switch(config)# interface vlan 50
switch(config-if)# no autostate
switch(config)# copy running-config startup-config
```

This example shows how to enable the systems autostate configuration:

```
switch(config)# show interface-vlan 2
Vlan2 is down, line protocol is down, autostate enabled
Hardware is EtherSVI, address is 547f.ee40.a17c
MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec
```

Configuring a Default Interface

The default interface feature allows you to clear the existing configuration of multiple interfaces such as Ethernet, loopback, management, VLAN, and port-channel interfaces. All user configuration under a specified interface will be deleted.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **default interface** *type interface number*
3. switch(config)# **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **default interface** *type interface number*	Deletes the configuration of the interface and restores the default configuration. The following are the supported interfaces: • **ethernet** • **loopback** • **mgmt** • **port-channel** • **vlan**
Step 3	switch(config)# **exit**	Exits global configuration mode.

Example

This example shows how to delete the configuration of an Ethernet interface and revert it to its default configuration:

```
switch# configure terminal
switch(config)# default interface ethernet 1/3
.......Done
switch(config)# exit
```

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
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	(Optional) switch(config)# [**no**] **cdp advertise** {**v1** | **v2** }	Configures the version to use to send CDP advertisements. Version-2 is the default state. Use the **no** form of the command to return to its default setting.
Step 3	(Optional) switch(config)# [**no**] **cdp format device-id** {**mac-address** | **serial-number** | **system-name**}	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 **no** form of the command to return to its default setting.
Step 4	(Optional) switch(config)# [**no**] **cdp holdtime** *seconds*	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 **no** form of the command to return to its default setting.
Step 5	(Optional) switch(config)# [**no**] **cdp timer** *seconds*	Sets the transmission frequency of CDP updates in seconds. The range is 5 to 254; the default is 60 seconds. Use the **no** form of the command to return to its default setting.

Example

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

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

Example

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

Example

This example shows how to enable 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* | *loopback*}
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	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **errdisable recovery cause** {*all* | *udld* | *bpduguard* | *link-flap* | *failed-port-state* | *pause-rate-limit* | *loopback*}	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.
Step 3	switch(config)# **show interface status err-disabled**	Displays information about err-disabled interfaces.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

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

```
switch# configure terminal
switch(config)# errdisable recovery cause loopback
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	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **errdisable recovery interval** *interval*	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	switch(config)# **show interface status err-disabled**	Displays information about err-disabled interfaces.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable 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
```

Disabling the Error-Disabled Recovery

You can disable recovery of an interface from the err-disabled state.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **no errdisable recovery cause** {*all* | *udld* | *bpduguard* | *link-flap* | *failed-port-state* | *pause-rate-limit* | *loopback*}
3. (Optional) switch(config)# **show interface status err-disabled**
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no errdisable recovery cause** {*all* | *udld* | *bpduguard* | *link-flap* | *failed-port-state* | *pause-rate-limit* | *loopback*}	Specifies a condition under which the interface reverts back to the default err-disabled state.
Step 3	(Optional) switch(config)# **show interface status err-disabled**	Displays information about err-disabled interfaces.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to disable err-disabled recovery:

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

Configuring the Debounce Timer

You can enable the debounce timer for Ethernet ports by specifying a debounce time, in milliseconds (ms), or disable the timer by specifying a debounce time of 0. By default, the debounce timer is set to 100 ms, which results in the debounce timer not running.

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

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

Example

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

```
switch# configure terminal
switch(config)# interface ethernet 3/1
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 3/1
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# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Enters interface configuration mode for the specified interface.
Step 3	switch(config-if)# **description** *test*	Specifies the description for the interface.

Example

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.

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
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Enters interface configuration mode for the specified interface.
Step 3	switch(config-if)# **shutdown**	Disables the interface.

	Command or Action	Purpose
Step 4	switch(config-if)# **no shutdown**	Restarts the interface.

Example

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
```

Configuring MAC addresses Limitation on a VLAN

Cisco Nexus 3600 Series switches provides the ability to set an upper limit for the number of MAC addresses that can reside inside MAC address table of a Line-card Expansion-module (LEM). You can configure the limitations at System, VLAN, port, trunk and tunnel levels. For instance if the specified VLAN limitation is 2000 MACs, the Layer 2 Forwarding Manager (L2FM) accepts the first 2000 MACs it receives and reject the remaining MACs. To configure MAC address limitation on VLAN, complete the following steps:

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **mac address-table limit system** *value*
3. switch(config)# **mac address-table limit vlan** *value*
4. switch(config)# **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **mac address-table limit system** *value*	Specifies an upper limit for MAC learning at system level.
Step 3	switch(config)# **mac address-table limit vlan** *value*	Specifies an upper limit for MAC learning at VLAN level.
Step 4	switch(config)# **exit**	Exits configuration mode.

Example

This example shows how to configure the upper limit for MAC learning at system and VLAN levels:

```
switch# configure terminal
```

```
switch(config)# mac address-table limit system 10000
Configuring Mac address limit will result in flushing existing Macs in the specified
VLAN/System.Proceed(yes/no)? [no] yes
switch(config)# mac address-table limit vlan 30 3000
Configuring Mac address limit will result in flushing existing Macs in the specified
VLAN/System.Proceed(yes/no)? [no] yes
switch(config)# exit
```

This example shows how to display the MAC address limitations:

```
switch# configure terminal
switch(config)# sh mac address-table limit

System Limit: 10000


Vlan    Learning Limit
----    ----------
1       196000
20      196000
30      3000
100     196000
switch(config)# exit
```

Configuring Custom EtherType or Tag Protocol Identifier (TPID)

The switch uses a default ethertype of 0x8100 for 802.1Q and Q-in-Q encapsulations. You can configure EtherTypes 0x9100, 0x9200 and 0x88a8 on a per port basis by enabling the **dot1q ethertype** command on the switchport interface. You can configure a custom EtherType field value on a port to support network devices that do not use the standard 0x8100 EtherType field value on 802.1Q-tagged or 802.1p-tagged frames.

Note You must set the EtherType or TPID only on the egress trunk interface that carries double tagged frames. EtherType value impacts all the tagged packets that go out on the interface (on both Q-in-Q and 802.1Q packets).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface ethernet** *slot/port*
3. switch(config-if)# **switchport**
4. switch(config-if)# **switchport mode**
5. switch(config-if)# **switchport dot1q ethertype** *value*
6. (Optional) switch(config-if)# **switchport access vlan** *value*
7. switch(config-if)# **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **interface ethernet** *slot/port*	Specifies an interface to configure, and enters interface configuration mode.
Step 3	switch(config-if)# **switchport**	Sets the interface as a Layer 2 switching port.
Step 4	switch(config-if)# **switchport mode**	Sets the interface as a Layer 2 switching port mode.
Step 5	switch(config-if)# **switchport dot1q ethertype** *value*	Sets the EtherType for the Q-in-Q tunnel on the port.
Step 6	(Optional) switch(config-if)# **switchport access vlan** *value*	Sets the interface access VLAN.
Step 7	switch(config-if)# **exit**	Exits configuration mode.

Example

This example shows how to configure custom ethertype on an 802.1Q tunnel port:

```
switch# configure terminal
switch(config)# interface ethernet 1/1
switch(config-if)# switchport
switch(config-if)# switchport mode dot1q-tunnel
switch(config-if)# switchport dot1q ethertype 0x9100
switch(config-if)# switchport access vlan 30
switch(config-if)# exit
switch(config)# exit
```

Configuring Downlink Delay

You can operationally enable uplink SFP+ ports before downlink RJ-45 ports after a reload on a Cisco Nexus 3048 switch by delaying enabling the RJ-45 ports in the hardware until the SFP+ ports are enabled.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **downlink delay enable** | **disable** [**timeout** *time-out*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **downlink delay enable** | **disable** [**timeout** *time-out*]	Enables or disables downlink delay and configures the timeout.

Example

This example shows how to enable downlink delay and configure the delay timeout on the switch:

```
switch# configure terminal
switch(config)# downlink delay enable timeout 45
```

Displaying Interface Information

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

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

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
  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 CDP neighbors:

```
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
d13-dist-1               mgmt0           148     S I      WS-C2960-24TC  Fas0/9
n5k(FLC12080012)         Eth1/5          8       S I s    N5K-C5020P-BA  Eth1/5
```

CHAPTER 4

Configuring Layer 3 Interfaces

- Information About Layer 3 Interfaces, on page 35
- Routed Interfaces, on page 35
- Subinterfaces, on page 36
- VLAN Interfaces, on page 37
- Changing VRF Membership for an Interface, on page 38
- Notes About Changing VRF Membership for an Interface, on page 38
- Loopback Interfaces, on page 39
- IP Unnumbered, on page 39
- Tunnel Interfaces, on page 39
- Guidelines and Limitations for Layer 3 Interfaces, on page 39
- Default Settings for Layer 3 Interfaces, on page 40
- SVI Autostate Disable, on page 40
- Configuring Layer 3 Interfaces, on page 40
- Verifying the Layer 3 Interfaces Configuration, on page 52
- Monitoring Layer 3 Interfaces, on page 53
- Configuration Examples for Layer 3 Interfaces, on page 55
- Related Documents for Layer 3 Interfaces, on page 55

Information About Layer 3 Interfaces

Layer 3 interfaces forward packets to another device using static or dynamic routing protocols. You can use Layer 3 interfaces for IP routing and inter-VLAN routing of Layer 2 traffic.

Routed Interfaces

You can configure a port as a Layer 2 interface or a Layer 3 interface. A routed interface is a physical port that can route IP traffic to another device. A routed interface is a Layer 3 interface only and does not support Layer 2 protocols, such as the Spanning Tree Protocol (STP).

All Ethernet ports are Layer 2 (switchports) by default. You can change this default behavior using the **no switchport** command from interface configuration mode. To change multiple ports at one time, you can specify a range of interfaces and then apply the **no switchport** command.

You can assign an IP address to the port, enable routing, and assign routing protocol characteristics to this routed interface.

You can assign a static MAC address to a Layer 3 interface. The default MAC address for a Layer 3 interface is the MAC address of the virtual device context (VDC) that is associated with it. You can change the default MAC address of the Layer 3 interface by using the **mac-address** command from the interface configuration mode. A static MAC address can be configured on SVI, Layer 3 interfaces, port channels, Layer 3 subinterfaces, and tunnel interfaces. You can also configure static MAC addresses on a range of ports and port channels. However, all ports must be in Layer 3. Even if one port in the range of ports is in Layer 2, the command is rejected and an error message appears. For information on configuring MAC addresses, see the Layer 2 Switching Configuration Guide for your device.

You can also create a Layer 3 port channel from routed interfaces.

Routed interfaces and subinterfaces support exponentially decayed rate counters. Cisco NX-OS tracks the following statistics with these averaging counters:

- Input packets/sec
- Output packets/sec
- Input bytes/sec
- Output bytes/sec

Subinterfaces

You can create virtual subinterfaces on a parent interface configured as a Layer 3 interface. A parent interface can be a physical port or a port channel.

Subinterfaces divide the parent interface into two or more virtual interfaces on which you can assign unique Layer 3 parameters such as IP addresses and dynamic routing protocols. The IP address for each subinterface should be in a different subnet from any other subinterface on the parent interface.

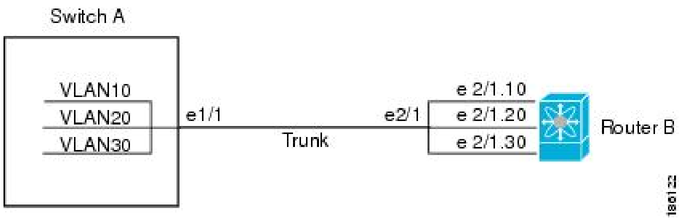
You create a subinterface with a name that consists of the parent interface name (for example, Ethernet 2/1) followed by a period and then by a number that is unique for that subinterface. For example, you could create a subinterface for Ethernet interface 2/1 named Ethernet 2/1.1 where .1 indicates the subinterface.

Cisco NX-OS enables subinterfaces when the parent interface is enabled. You can shut down a subinterface independent of shutting down the parent interface. If you shut down the parent interface, Cisco NX-OS shuts down all associated subinterfaces as well.

One use of subinterfaces is to provide unique Layer 3 interfaces to each VLAN that is supported by the parent interface. In this scenario, the parent interface connects to a Layer 2 trunking port on another device. You configure a subinterface and associate the subinterface to a VLAN ID using 802.1Q trunking.

The following figure shows a trunking port from a switch that connects to router B on interface E 2/1. This interface contains three subinterfaces that are associated with each of the three VLANs that are carried by the trunking port.

Figure 2: Subinterfaces for VLANs



VLAN Interfaces

A VLAN interface or a switch virtual interface (SVI) is a virtual routed interface that connects a VLAN on the device to the Layer 3 router engine on the same device. Only one VLAN interface can be associated with a VLAN, but you need to configure a VLAN interface for a VLAN only when you want to route between VLANs or to provide IP host connectivity to the device through a virtual routing and forwarding (VRF) instance that is not the management VRF. When you enable VLAN interface creation, Cisco NX-OS creates a VLAN interface for the default VLAN (VLAN 1) to permit remote switch administration.

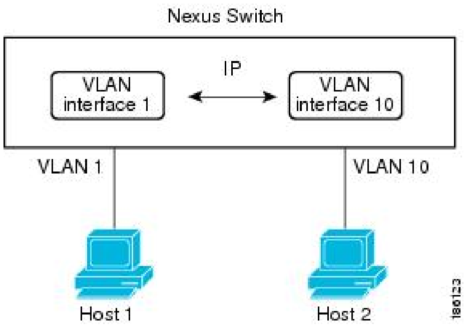
You must enable the VLAN network interface feature before you can configure it. The system automatically takes a checkpoint prior to disabling the feature, and you can roll back to this checkpoint. For information about rollbacks and checkpoints, see the System Management Configuration Guide for your device.

Note You cannot delete the VLAN interface for VLAN 1.

You can route across VLAN interfaces to provide Layer 3 inter-VLAN routing by configuring a VLAN interface for each VLAN that you want to route traffic to and assigning an IP address on the VLAN interface. For more information on IP addresses and IP routing, see the Unicast Routing Configuration Guide for your device.

The following figure shows two hosts connected to two VLANs on a device. You can configure VLAN interfaces for each VLAN that allows Host 1 to communicate with Host 2 using IP routing between the VLANs. VLAN 1 communicates at Layer 3 over VLAN interface 1and VLAN 10 communicates at Layer 3 over VLAN interface 10.

Figure 3: Connecting Two VLANs with VLAN Interfaces



Changing VRF Membership for an Interface

When you enter the **vrf member** command under an interface, you receive an alert regarding the deletion of interface configurations and to notify the clients/listeners (such as CLI-Server) to delete configurations with respect to the interface.

Entering the **system vrf-member-change retain-l3-config** command enables the retention of the Layer 3 configuration when the VRF member changes on the interface. It does this by sending notification to the clients/listeners to store (buffer) the existing configurations, delete the configurations from the old vrf context, and reapply the stored configurations under the new VRF context.

Note When the **system vrf-member-change retain-l3-config** command is enabled, the Layer 3 configuration is not deleted and remains stored (buffered). When this command is not enabled (default mode), the Layer 3 configuration is not retained when the VRF member changes.

You can disable the retention of the Layer 3 configuration with the **no system vrf-member-change retain-l3-config** command. In this mode, the Layer 3 configuration is not retained when the VRF member changes.

Notes About Changing VRF Membership for an Interface

- Momentary traffic loss may occur when changing the VRF name.
- Only the configurations under the interface level are processed when the **system vrf-member-change retain-l3-config** command is enabled. You must manually process any configurations at the router level to accommodate routing protocols after a VRF change.
- The **system vrf-member-change retain-l3-config** command supports interface level configurations with:
    - Layer 3 configurations maintained by the CLI Server, such as **ip address** and **ipv6 address** (secondary) and all OSPF/ISIS/EIGRP CLIs available under the interface configuration.
    - HSRP
    - DHCP Relay Agent CLIs, such as **ip dhcp relay address** [**use-vrf**] and **ipv6 dhcp relay address** [**use-vrf**].
- For DHCP:
    - As a best practice, the client and server interface VRF should be changed one at a time. Otherwise, the DHCP packets cannot be exchanged on the relay agent.
    - When the client and server are in different VRFs, use the **ip dhcp relay address** [**use-vrf**] command to exchange the DHCP packets in the relay agent over the different VRFs.

Loopback Interfaces

A loopback interface is a virtual interface with a single endpoint that is always up. Any packet that is transmitted over a loopback interface is immediately received by this interface. Loopback interfaces emulate a physical interface.

You can use loopback interfaces for performance analysis, testing, and local communications. Loopback interfaces can act as a termination address for routing protocol sessions. This loopback configuration allows routing protocol sessions to stay up even if some of the outbound interfaces are down.

IP Unnumbered

The IP unnumbered feature enables the processing of IP packets on a point to point (p2p) interface without explicitly configuring a unique IP address on it. This approach borrows an IP address from another interface and conserves address space on point to point links.

Any interface which conforms to the point to point mode can be used as an IP unnumbered interface. The IP unnumbered feature is supported only on Ethernet interfaces and sub-interfaces. The borrowed interface can only be a loopback interface and is known as the numbered interface.

A loopback interface is ideal as a numbered interface in that it is always functionally up. However, because loopback interfaces are local to a switch/router, the reachability of unnumbered interfaces first needs to be established through static routes or by using an interior gateway protocol, such as OSPF or ISIS.

IP unnumbered feature is supported on port channel interfaces and sub-interfaces. The borrowed interface can only be a loopback interface and is known as the numbered interface.

Tunnel Interfaces

Cisco NX-OS supports tunnel interfaces as IP tunnels. IP tunnels can encapsulate a same- layer or higher layer protocol and transport the result over IP through a tunnel that is created between two routers.

Note IP-in-IP tunnel encapsulation and decapsulation is not supported on Cisco Nexus N3K-C36180YC-R platform switches.

Guidelines and Limitations for Layer 3 Interfaces

Layer 3 interfaces have the following configuration guidelines and limitations:

- VLAN/SVI is not removed from the Layer 3 interface table, after the configuration is removed. The VLAN itself should be removed from the Layer 3 interface table.
- If you change a Layer 3 interface to a Layer 2 interface, Cisco NX-OS shuts down the interface, reenables the interface, and removes all configuration specific to Layer 3.

- If you change a Layer 2 interface to a Layer 3 interface, Cisco NX-OS shuts down the interface, reenables the interface, and deletes all configuration specific to Layer 2.

Default Settings for Layer 3 Interfaces

The default setting for the Layer 3 Admin state is Shut.

SVI Autostate Disable

The SVI Autostate Disable feature enables the Switch Virtual Interface (SVI) to be in the “up” state even if no interface is in the “up” state in the corresponding VLAN.

An SVI is also a virtual routed interface that connects a VLAN on the device to the Layer 3 router engine on the same device. The ports in a VLAN determine the operational state of the corresponding SVI. An SVI interface on a VLAN comes “up” when at least one port in the corresponding VLAN is in the Spanning Tree Protocol (STP) forwarding state. Similarly, the SVI interface goes “down” when the last STP forwarding port goes down or to any other state. This characteristic of SVI is called 'Autostate'.

You can create SVIs to define Layer 2 or Layer 3 boundaries on VLANs, or use the SVI interface to manage devices. In the second scenario, the SVI Autostate Disable feature ensures that the SVI interface is in the “up” state even if no interface is in the “up” state in the corresponding VLAN.

Configuring Layer 3 Interfaces

Configuring a Routed Interface

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface ethernet** *slot/port*
3. switch(conifg-if)# **no switchport**
4. switch(config-if)# [**ip**|**ipv6**]*ip-address/length*
5. (Optional) switch(config-if)# **medium** {**broadcast** | **p2p**}
6. (Optional) switch(config-if)# **show interfaces**
7. (Optional) switch(config-if)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port*	Enters interface configuration mode.
Step 3	switch(conifg-if)# **no switchport**	Configures the interface as a Layer 3 interface and deletes any configuration specific to Layer 2 on this interface.

	Command or Action	Purpose
		Note To convert a Layer 3 interface back into a Layer 2 interface, use the **switchport** command.
Step 4	switch(config-if)# [**ip**|**ipv6**]*ip-address/length*	Configures an IP address for this interface.
Step 5	(Optional) switch(config-if)# **medium** {**broadcast** | **p2p**}	Configures the interface medium as either point to point or broadcast. **Note** The default setting is broadcast, and this setting does not appear in any of the **show** commands. However, if you do change the setting to **p2p**, you will see this setting when you enter the **show running-config** command.
Step 6	(Optional) switch(config-if)# **show interfaces**	Displays the Layer 3 interface statistics.
Step 7	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure an IPv4-routed Layer 3 interface:

```
switch# configure terminal
switch(config)# interface ethernet 2/1
switch(config-if)# no switchport
switch(config-if)# ip address 192.0.2.1/8
switch(config-if)# copy running-config startup-config
```

Configuring a Subinterface

Before you begin

- Configure the parent interface as a routed interface.
- Create the port-channel interface if you want to create a subinterface on that port channel.

SUMMARY STEPS

1. (Optional) switch(config-if)# **copy running-config startup-config**
2. switch(config)# **interface ethernet** *slot/port***.***number*
3. switch(config-if)# [**ip** | **ipv6**] **address** *ip-address/length*
4. switch(config-if)# **encapsulation dot1Q** *vlan-id*
5. (Optional) switch(config-if)# **show interfaces**
6. (Optional) switch(config-if)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 2	switch(config)# **interface ethernet** *slot/port***.***number*	Enters interface configuration mode. The range for the *slot* is from 1 to 255. The range for the *port* is from 1 to 128.
Step 3	switch(config-if)# [**ip** | **ipv6**] **address** *ip-address/length*	Configures an IP address for this interface.
Step 4	switch(config-if)# **encapsulation dot1Q** *vlan-id*	Configures IEEE 802.1Q VLAN encapsulation on the subinterface. The range for the *vlan-id* is from 2 to 4093.
Step 5	(Optional) switch(config-if)# **show interfaces**	Displays the Layer 3 interface statistics.
Step 6	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to create a subinterface:

```
switch# configure terminal
switch(config)# interface ethernet 2/1
switch(config-if)# ip address 192.0.2.1/8
switch(config-if)# encapsulation dot1Q 33
switch(config-if)# copy running-config startup-config
```

Configuring the Bandwidth on an Interface

You can configure the bandwidth for a routed interface, port channel, or subinterface.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface ethernet** *slot/port*
3. switch(conifg-if)# **bandwidth** [*value* | **inherit** [*value*]]
4. (Optional) switch(config-if)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port*	Enters interface configuration mode. The range for the *slot* is from 1 to 255. The range for the *port* is from 1 to 128.

	Command or Action	Purpose
Step 3	switch(conifg-if)# **bandwidth** [*value* | **inherit** [*value*]]	Configures the bandwidth parameter for a routed interface, port channel, or subinterface, as follows: • *value*—Size of the bandwidth in kilobytes. The range is from 1 to 10000000. • **inherit**—Indicates that all subinterfaces of this interface inherit either the bandwidth value (if a value is specified) or the bandwidth of the parent interface (if a value is not specified).
Step 4	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure Ethernet interface 2/1 with a bandwidth value of 80000:

```
switch# configure terminal
switch(config)# interface ethernet 2/1
switch(config-if)# bandwidth 80000
switch(config-if)# copy running-config startup-config
```

Configuring a VLAN Interface

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **feature interface-vlan**
3. switch(config)# **interface vlan** *number*
4. switch(config-if)# [**ip** | **ipv6**] **address** *ip-address/length*
5. switch(config-if)# **no shutdown**
6. (Optional) switch(config-if)# **show interface vlan** *number*
7. (Optional) switch(config-if)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature interface-vlan**	Enables VLAN interface mode.
Step 3	switch(config)# **interface vlan** *number*	Creates a VLAN interface. The *number* range is from 1 to 4094.
Step 4	switch(config-if)# [**ip** | **ipv6**] **address** *ip-address/length*	Configures an IP address for this interface.

	Command or Action	Purpose
Step 5	switch(config-if)# **no shutdown**	Brings the interface up administratively.
Step 6	(Optional) switch(config-if)# **show interface vlan** *number*	Displays the VLAN interface statistics. The *number* range is from 1 to 4094.
Step 7	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to create a VLAN interface:

```
switch# configure terminal
switch(config)# feature interface-vlan
switch(config)# interface vlan 10
switch(config-if)# ip address 192.0.2.1/8
switch(config-if)# copy running-config startup-config
```

Enabling Layer 3 Retention During VRF Membership Change

The following steps enable the retention of the Layer 3 configuration when changing the VRF membership on the interface.

SUMMARY STEPS

1. **configure terminal**
2. **system vrf-member-change retain-l3-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**system vrf-member-change retain-l3-config**
	Example:
	`switch(config)# system vrf-member-change retain-l3-config`
	`Warning: Will retain L3 configuration when vrf member change on interface.`
		Enables Layer 3 configuration retention during VRF membership change.
		Note To disable the retention of the Layer 3 configuration, use the **no system vrf-member-change retain-l3-config** command.

Configuring a Loopback Interface

Before you begin

Ensure that the IP address of the loopback interface is unique across all routers on the network.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface loopback** *instance*
3. switch(config-if)# [**ip** | **ipv6**] **address** *ip-address/length*
4. (Optional) switch(config-if)# **show interface loopback** *instance*
5. (Optional) switch(config-if)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface loopback** *instance*	Creates a loopback interface. The *instance* range is from 0 to 1023.
Step 3	switch(config-if)# [**ip** | **ipv6**] **address** *ip-address/length*	Configures an IP address for this interface.
Step 4	(Optional) switch(config-if)# **show interface loopback** *instance*	Displays the loopback interface statistics. The *instance* range is from 0 to 1023.
Step 5	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to create a loopback interface:

```
switch# configure terminal
switch(config)# interface loopback 0
switch(config-if)# ip address 192.0.2.100/8
switch(config-if)# copy running-config startup-config
```

Configuring IP Unnumbered on an Ethernet Interface

You can configure the IP unnumbered feature on an ethernet interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot/port* **port-channel**
3. **medium p2p**

4. **ip unnumbered** *type number*

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface ethernet** *slot/port* **port-channel**
	Example:
	`switch(config)# interface ethernet 1/1`
	`switch(config-if)#`
	`switch(config)# interface port-channel 1/1`
	`switch(config-if)#`
		Enters interface configuration mode. Supports Ethernet and Port-channel

Step 3	**medium p2p** **Example:** `switch(config-if)# medium p2p`	Configures the interface medium as point to point.
Step 4	**ip unnumbered** *type number* **Example:** `switch(config-if)# ip unnumbered loopback 100`	Enables IP processing on an interface without assigning an explicit IP address to the interface. *type* and *number* specify another interface on which the router has an assigned IP address. The interface specified cannot be another unnumbered interface. **Note** *type* is limited to **loopback**. (7.0(3)I3(1) and later)

Assigning an Interface to a VRF

Before you begin

Assign the IP address for a tunnel interface after you have configured the interface for a VRF.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *interface-typenumber*
3. switch(conifg-if)#**vrf member** *vrf-name*
4. switch(config-if)# FID cleanup[**ip** | **ipv6**]*ip-address/length*
5. (Optional) switch(config-if)# **show vrf** [*vrf-name*] **interface** *interface-type number*
6. (Optional) switch(config-if)# **show interfaces**
7. (Optional) switch(config-if)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-typenumber*	Enters interface configuration mode.
Step 3	switch(conifg-if)#**vrf member** *vrf-name*	Adds this interface to a VRF.
Step 4	switch(config-if)# FID cleanup[**ip** | **ipv6**]*ip-address/length*	Configures an IP address for this interface. You must do this step after you assign this interface to a VRF.
Step 5	(Optional) switch(config-if)# **show vrf** [*vrf-name*] **interface** *interface-type number*	Displays VRF information.
Step 6	(Optional) switch(config-if)# **show interfaces**	Displays the Layer 3 interface statistics.
Step 7	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to add a Layer 3 interface to the VRF:

```
switch# configure terminal
switch(config)# interface loopback 0
switch(config-if)# vrf member RemoteOfficeVRF
switch(config-if)# ip address 209.0.2.1/16
switch(config-if)# copy running-config startup-config
```

Configuring an Interface MAC Address

You can configure a static MAC address on SVI, Layer 3 interfaces, port channels, Layer 3 subinterfaces, and tunnel interfaces. You can also configure static MAC addresses on a range of ports and port channels. However, all ports must be in Layer 3. Even if one port in the range of ports is in Layer 2, the command is rejected and an error message appears.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface ethernet** *slot/port*
3. switch(config-if)# [**no**] **mac-address** *static router MAC address*
4. switch(config-if)# **show interface ethernet** *slot/port*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *slot/port*	Enters interface configuration mode.

	Command or Action	Purpose
Step 3	switch(config-if)# [**no**] **mac-address** *static router MAC address*	Configures the interface MAC address. The **no** form removes the configuration. You can enter the MAC address in any one of the four supported formats: • E.E.E • EE-EE-EE-EE-EE-EE • EE:EE:EE:EE:EE:EE • EEEE.EEEE.EEEE Do not enter any of the following invalid MAC addresses: • Null MAC address—0000.0000.0000 • Broadcast MAC address—FFFF.FFFF.FFFF • Multicast MAC address—0100.DAAA.ADDD
Step 4	switch(config-if)# **show interface ethernet** *slot/port*	(Optional) Displays all information for the interface.

Example

This example shows how to configure an interface MAC address:

```
switch# configure terminal
switch(config)# interface ethernet 3/3
switch(config-if)# mac-address aaaa.bbbb.dddd
switch(config-if)# show interface ethernet 3/3
switch(config-if)#
```

Configuring a MAC-Embedded IPv6 Address

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type slot/port*
3. switch(config-if)# **no switchport**
4. switch(config-if)# **mac-address ipv6-extract**
5. switch(config-if)# **ipv6 address** *ip-address/length*
6. switch(config-if)# **ipv6 nd mac-extract** [**exclude nud-phase**]
7. (Optional) switch(config)# **show ipv6 icmp interface** *type slot/port*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot/port*	Enters the interface configuration mode for the specified interface.

	Command or Action	Purpose
Step 3	switch(config-if)# **no switchport**	Configures the interface as a Layer 3 interface and deletes any configuration specific to Layer 2 on this interface. **Note** To convert a Layer 3 interface back into a Layer 2 interface, use the **switchport** command.
Step 4	switch(config-if)# **mac-address ipv6-extract**	Extracts the MAC address embedded in the IPv6 address configured on the interface. **Note** The MEv6 configuration is currently not supported with the EUI-64 format of IPv6 address.
Step 5	switch(config-if)# **ipv6 address** *ip-address/length*	Configures an IPv6 address for this interface.
Step 6	switch(config-if)# **ipv6 nd mac-extract** [**exclude nud-phase**]	Extracts the next-hop MAC address embedded in a next-hop IPv6 address. The **exclude nud-phase** option blocks packets during the ND phase only. When the **exclude nud-phase** option is not specified, packets are blocked during both ND and Neighbor Unreachability Detection (NUD) phases.
Step 7	(Optional) switch(config)# **show ipv6 icmp interface** *type slot/port*	Displays IPv6 Internet Control Message Protocol version 6 (ICMPv6) interface information.

Example

This example shows how to configure a MAC-embedded IPv6 address with ND mac-extract enabled:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# interface ethernet 1/3
switch(config-if)# no switchport
switch(config-if)# mac-address ipv6-extract
switch(config-if)# ipv6 address 2002:1::10/64
switch(config-if)# ipv6 nd mac-extract
switch(config-if)# show ipv6 icmp interface ethernet 1/3
ICMPv6 Interfaces for VRF "default"
Ethernet1/3, Interface status: protocol-up/link-up/admin-up
  IPv6 address: 2002:1::10
  IPv6 subnet:  2002:1::/64
  IPv6 interface DAD state:  VALID
  ND mac-extract : Enabled
  ICMPv6 active timers:
      Last Neighbor-Solicitation sent: 00:01:39
      Last Neighbor-Advertisement sent: 00:01:40
      Last Router-Advertisement sent: 00:01:41
      Next Router-Advertisement sent in: 00:03:34
  Router-Advertisement parameters:
      Periodic interval: 200 to 600 seconds
      Send "Managed Address Configuration" flag: false
      Send "Other Stateful Configuration" flag: false
      Send "Current Hop Limit" field: 64
```

```
        Send "MTU" option value: 1500
        Send "Router Lifetime" field: 1800 secs
        Send "Reachable Time" field: 0 ms
        Send "Retrans Timer" field: 0 ms
        Suppress RA: Disabled
        Suppress MTU in RA: Disabled
    Neighbor-Solicitation parameters:
        NS retransmit interval: 1000 ms
    ICMPv6 error message parameters:
        Send redirects: true
        Send unreachables: false
    ICMPv6-nd Statisitcs (sent/received):
        RAs: 3/0, RSs: 0/0, NAs: 2/0, NSs: 7/0, RDs: 0/0
        Interface statistics last reset: never
switch(config)#
```

This example shows how to configure a MAC-embedded IPv6 address with ND mac-extract (excluding NUD phase) enabled:

```
switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
switch(config)# interface ethernet 1/5
switch(config-if)# no switchport
switch(config-if)# mac-address ipv6-extract
switch(config-if)# ipv6 address 2002:2::10/64
switch(config-if)# ipv6 nd mac-extract exclude nud-phase
switch(config-if)# show ipv6 icmp interface ethernet 1/5
ICMPv6 Interfaces for VRF "default"
Ethernet1/5, Interface status: protocol-up/link-up/admin-up
  IPv6 address: 2002:2::10
  IPv6 subnet:  2002:2::/64
  IPv6 interface DAD state:  VALID
  ND mac-extract : Enabled (Excluding NUD Phase)
  ICMPv6 active timers:
        Last Neighbor-Solicitation sent: 00:06:45
        Last Neighbor-Advertisement sent: 00:06:46
        Last Router-Advertisement sent: 00:02:18
        Next Router-Advertisement sent in: 00:02:24
  Router-Advertisement parameters:
        Periodic interval: 200 to 600 seconds
        Send "Managed Address Configuration" flag: false
        Send "Other Stateful Configuration" flag: false
        Send "Current Hop Limit" field: 64
        Send "MTU" option value: 1500
        Send "Router Lifetime" field: 1800 secs
        Send "Reachable Time" field: 0 ms
        Send "Retrans Timer" field: 0 ms
        Suppress RA: Disabled
        Suppress MTU in RA: Disabled
  Neighbor-Solicitation parameters:
        NS retransmit interval: 1000 ms
  ICMPv6 error message parameters:
        Send redirects: true
        Send unreachables: false
  ICMPv6-nd Statisitcs (sent/received):
        RAs: 6/0, RSs: 0/0, NAs: 2/0, NSs: 7/0, RDs: 0/0
        Interface statistics last reset: never
switch(config-if)#
```

Configuring SVI Autostate Disable

You can configure a SVI to remain active even if no interfaces are up in the corresponding VLAN. This enhancement is called Autostate Disable.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **[no] system default interface-vlan autostate**
3. switch(config)# **feature interface-vlan**
4. switch(config)# **interface vlan** *vlan id*
5. (config-if)# [**no**] **autostate**
6. (config-if)# **end**
7. **show running-config interface vlan** *vlan id*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **[no] system default interface-vlan autostate**	Reenables the system default autostate behavior on Switching Virtual Interface (SVI) in a VLAN. Use the no form of the command to disable the autostate behavior on SVI.
Step 3	switch(config)# **feature interface-vlan**	Enables the creation of VLAN interfaces SVI.
Step 4	switch(config)# **interface vlan** *vlan id*	Disables the VLAN interface and enters interface configuration mode.
Step 5	(config-if)# [**no**] **autostate**	Disables the default autostate behavior of SVIs on the VLAN interface.
Step 6	(config-if)# **end**	Returns to privileged EXEC mode.
Step 7	**show running-config interface vlan** *vlan id*	(Optional) Displays the running configuration for a specific port channel.

Example

This example shows how to configure the SVI Autostate Disable feature:

```
switch# configure terminal
switch(config)# system default interface-vlan autostate
switch(config)# feature interface-vlan
switch(config)# interface vlan 2
switch(config-if)# no autostate
switch(config-if)# end
```

Configuring a DHCP Client on an Interface

You can configure the IP address of a DHCP client on an SVI, a management interface, or a physical Ethernet interface.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface ethernet** *type slot/port* | **mgmt** *mgmt-interface-number* | **vlan** *vlan id*
3. switch(config-if)# [**no**] **ip** | **ipv6 address dhcp**
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *type slot/port* | **mgmt** *mgmt-interface-number* | **vlan** *vlan id*	Creates a physical Ethernet interface, a management interface, or a VLAN interface. The range of *vlan id* is from 1 to 4094.
Step 3	switch(config-if)# [**no**] **ip** | **ipv6 address dhcp**	Requests the DHCP server for an IPv4 or IPv6 address. The **no** form of this command removes any address that was acquired.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure the IP address of a DHCP client on an SVI:

```
switch# configure terminal
switch(config)# interface vlan 15
switch(config-if)# ip address dhcp
```

This example shows how to configure an IPv6 address of a DHCP client on a management interface:

```
switch# configure terminal
switch(config)# interface mgmt 0
switch(config-if)# ipv6 address dhcp
```

Verifying the Layer 3 Interfaces Configuration

Use one of the following commands to verify the configuration:

Command	Purpose
show interface ethernet *slot/port*	Displays the Layer 3 interface configuration, status, and counters (including the 5-minute exponentially decayed moving average of inbound and outbound packet and byte rates).
show interface ethernet *slot/port* **brief**	Displays the Layer 3 interface operational status.
show interface ethernet *slot/port* **capabilities**	Displays the Layer 3 interface capabilities, including port type, speed, and duplex.
show interface ethernet *slot/port* **description**	Displays the Layer 3 interface description.
show interface ethernet *slot/port* **status**	Displays the Layer 3 interface administrative status, port mode, speed, and duplex.
show interface ethernet *slot/port***.***number*	Displays the subinterface configuration, status, and counters (including the f-minute exponentially decayed moving average of inbound and outbound packet and byte rates).
show interface port-channel *channel-id***.***number*	Displays the port-channel subinterface configuration, status, and counters (including the 5-minute exponentially decayed moving average of inbound and outbound packet and byte rates).
show interface loopback *number*	Displays the loopback interface configuration, status, and counters.
show interface loopback *number* **brief**	Displays the loopback interface operational status.
show interface loopback *number* **description**	Displays the loopback interface description.
show interface loopback *number* **status**	Displays the loopback interface administrative status and protocol status.
show interface vlan *number*	Displays the VLAN interface configuration, status, and counters.
show interface vlan *number* **brief**	Displays the VLAN interface operational status.
show interface vlan *number* **description**	Displays the VLAN interface description.
show interface vlan *number* **status**	Displays the VLAN interface administrative status and protocol status.

Monitoring Layer 3 Interfaces

Use one of the following commands to display statistics about the feature:

Command	Purpose
load-interval *seconds* | **counter** {**1** | **2** | **3**} *seconds*	Sets three different sampling intervals to bit-rate and packet-rate statistics. The range is from 5 seconds to 300 seconds.
show interface ethernet *slot*/*port* **counters**	Displays the Layer 3 interface statistics (unicast, multicast, and broadcast).
show interface ethernet *slot*/*port* **counters brief** *load-interval-id*	Displays the Layer 3 interface input and output counters. The load interval ID specifies a single load interval ID to display the input and output rates. The load interval ID ranges between 1 and 3.
show interface ethernet *slot*/*port* **counters detailed** [**all**]	Displays the Layer 3 interface statistics. You can optionally include all 32-bit and 64-bit packet and byte counters (including errors).
show interface ethernet *slot*/*port* **counters error**	Displays the Layer 3 interface input and output errors.
show interface ethernet *slot*/*port* **counters snmp**	Displays the Layer 3 interface counters reported by SNMP MIBs. You cannot clear these counters.
show interface ethernet *slot*/*port***.***number* **counters**	Displays the subinterface statistics (unicast, multicast, and broadcast).
show interface port-channel *channel-id***.***number* **counters**	Displays the port-channel subinterface statistics (unicast, multicast, and broadcast).
show interface loopback *number* **counters**	Displays the loopback interface input and output counters (unicast, multicast, and broadcast).
show interface loopback *number* **counters detailed** [**all**]	Displays the loopback interface statistics. You can optionally include all 32-bit and 64-bit packet and byte counters (including errors).
show interface loopback *number* **counters errors**	Displays the loopback interface input and output errors.
show interface vlan *number* **counters**	Displays the VLAN interface input and output counters (unicast, multicast, and broadcast).
show interface vlan *number* **counters detailed** [*all*]	Displays the VLAN interface statistics. You can optionally include all Layer 3 packet and byte counters (unicast and multicast).
show interface vlan *counters* **snmp**	Displays the VLAN interface counters reported by SNMP MIBs. You cannot clear these counters.

Configuration Examples for Layer 3 Interfaces

This example shows how to configure Ethernet subinterfaces:

```
switch# configuration terminal
switch(config)# interface ethernet 2/1.10
switch(config-if)# description Layer 3 for VLAN 10
switch(config-if)# encapsulation dot1q 10
switch(config-if)# ip address 192.0.2.1/8
switch(config-if)# copy running-config startup-config
```

This example shows how to configure a VLAN interface:

```
switch# configuration terminal
switch(config)# interface vlan 100
switch(config-if)# copy running-config startup-config
```

This example shows how to configure Switching Virtual Interface (SVI) Autostate Disable:

```
switch# configure terminal
switch(config)# system default interface-vlan autostate
switch(config)# feature interface-vlan
switch(config)# interface vlan 2
switch(config-if)# no autostate
switch(config-if)# end
switch# show running-config interface vlan 2
```

This example shows how to configure a loopback interface:

```
switch# configuration terminal
switch(config)# interface loopback 3
switch(config-if)# no switchport
switch(config-if)# ip address 192.0.2.2/32
switch(config-if)# copy running-config startup-config
```

This example shows how to configure the three sample load intervals for an Ethernet port:

```
switch# configure terminal
switch(config)# interface ethernet 1/3
switch(config-if)# load-interval counter 1 5
switch(config-if)# load-interval counter 2 135
switch(config-if)# load-interval counter 3 225
switch(config-if)#
```

Related Documents for Layer 3 Interfaces

Related Topics	Document Title
Command syntax	*Cisco Nexus 3600 NX-OS Command Reference*
IP	“Configuring IP” chapter in the *Cisco Nexus 3600 NX-OS Unicast Routing Configuration Guide*

Related Topics	Document Title
VLAN	“Configuring VLANs” chapter in the *Cisco Nexus 3600 NX-OS Layer 2 Switching Configuration Guide*

CHAPTER 5

Configuring Port Channels

- Information About Port Channels, on page 57
- Understanding Port Channels, on page 58
- Compatibility Requirements, on page 58
- Load Balancing Using Port Channels, on page 60
- Guidelines and Limitations for ECMP, on page 61
- Symmetric Hashing, on page 62
- Understanding LACP, on page 62
- **Guidelines and Limitations**, on page 66
- Configuring Port Channels, on page 66
- Verifying Port Channel Configuration, on page 79
- Triggering the Port Channel Membership Consistency Checker, on page 79
- Verifying the Load-Balancing Outgoing Port ID , on page 80
- Port Profiles, on page 81
- Configuring Port Profiles, on page 82
- Creating a Port Profile, on page 82
- Entering Port-Profile Configuration Mode and Modifying a Port Profile, on page 84
- Assigning a Port Profile to a Range of Interfaces, on page 84
- Enabling a Specific Port Profile, on page 85
- Inheriting a Port Profile, on page 86
- Removing a Port Profile from a Range of Interfaces, on page 87
- Removing an Inherited Port Profile, on page 88

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. If the min-links configuration is > 1, the port channel will go down if the min-links condition is not met.

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 LACP, which is defined in IEEE 802.3ad. When you use LACP, the link passes protocol packets.

Related Topics

LACP Overview, on page 62

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 60.

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.

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
- 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.

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 6: Port Channel Load-Balancing Criteria

Configuration	Layer 2 Criteria	Layer 3 Criteria	Layer 4 Criteria
Destination MAC	Destination MAC	Destination MAC	Destination MAC

Configuration	Layer 2 Criteria	Layer 3 Criteria	Layer 4 Criteria
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
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

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.

Regardless of the load-balancing algorithm configured, multicast traffic uses the following methods for load balancing with port channels:

- Multicast traffic with Layer 4 information - Source IP address, source port, destination IP address, destination port
- Multicast traffic without Layer 4 information - Source IP address, destination IP address
- Non-IP multicast traffic - Source MAC address, destination MAC address

Note The hardware multicast hw-hash command is not supported on Cisco Nexus 3000 Series switches. It is recommended not to configure this command on these switches. By default, Cisco Nexus 3000 Series switches hash multicast traffic.

Guidelines and Limitations for ECMP

You might observe that load balancing with Layer 2/Layer 3 GW flows are not load balanced equally among all links when the switch comes up initially after reload. There are two CLIs to change the ECMP hash configuration in the hardware. The two CLI commands are mutually exclusive.

- Enter the **port-channel load-balance [src | src-dst | dst] mac** command for MAC-based only hash.
- For hash based on IP/Layer 4 ports, enter either the **ip load-share** or **port-channel load-balance** command.
- The **port-channel load-balance** command can overwrite the **ip load-share** command. It is better to enter the **port-channel load-balance** command which helps to set both the IP and MAC parameters.
- There are no options to force the hashing algorithm based on the IP/Layer 4 port. The default MAC configuration is always programmed as a part of the port channel configuration.

Symmetric Hashing

To be able to effectively monitor traffic on a port channel, it is essential that each interface connected to a port channel receives both forward and reverse traffic flows. Normally, there is no guarantee that the forward and reverse traffic flows will use the same physical interface. However, when you enable symmetric hashing on the port channel, bidirectional traffic is forced to use the same physical interface and each physical interface in the port channel is effectively mapped to a set of flows.

When symmetric hashing is enabled, the parameters used for hashing, such as the source and destination IP address, are normalized before they are entered into the hashing algorithm. This process ensures that when the parameters are reversed (the source on the forward traffic becomes the destination on the reverse traffic), the hash output is the same. Therefore, the same interface is chosen.

Symmetric hashing is supported only on Cisco Nexus 3600 Series switches.

Only the following load-balancing algorithms support symmetric hashing:

- source-dest-ip-only
- source-dest-port-only
- source-dest-ip
- source-dest-port
- source-dest-ip-gre

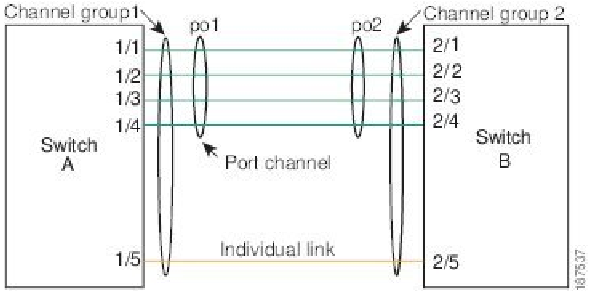
Understanding LACP

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 4: Individual Links Combined into a Port Channel



With LACP, just like with static port channels, you can bundle up to 32 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. The configuration could have an LACP configuration, like LACP min-links on a port channel, but with no members. In that case, you can disable LACP.

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. The **no lacp suspend-individual** configuration is supported by default on Cisco Nexus 3600 switches.

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: • Active • Passive	Can only be On.

LACP Port Channel Minimum Links and MaxBundle

A port channel aggregates similar ports to provide increased bandwidth in a single manageable interface. The introduction of the minimum links and MaxBundle feature further refines LACP port-channel operation and provides increased bandwidth in one manageable interface.

The LACP port channel MinLinks feature does the following:

- Configures the minimum number of port channel interfaces that must be linked and bundled in the LACP port channel.
- Prevents a low-bandwidth LACP port channel from becoming active.
- Causes the LACP port channel to become inactive if only a few active members ports supply the required minimum bandwidth.

The LACP MaxBundle defines the maximum number of bundled ports allowed in a LACP port channel. The LACP MaxBundle feature does the following:

- Defines an upper limit on the number of bundled ports in an LACP port channel.
- Allows hot-standby ports with fewer bundled ports. (For example, in an LACP port channel with five ports, you can designate two of those ports as hot-standby ports.)

Note The minimum links and maxbundle feature works only with LACP port channels. However, the device allows you to configure this feature in non-LACP port channels, but the feature is not operational.

Guidelines and Limitations

Port channeling has the following configuration guidelines and limitations:

- On a Cisco Nexus 36180YC switch, the first 24 ports are part of the same quadrant. Ports in the same quadrant must have same speed (1/10G or 25G) on all ports. Having different speed on ports in a quadrant is not supported. If you set different speed in any of the ports in a quadrant, ports go into error disable state. Interfaces in same quadrant are:
    - 1–4
    - 5–8
    - 9–12
    - 13–16
    - 17–20
    - 21–24
    - 25–28
    - 29–32
    - 33–36
    - 37–40
    - 41–44
    - 45–48

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.

Note Channel member ports cannot be a source or destination SPAN port.

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
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	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.
Step 3	switch(config)# **no interface port-channel** *channel-number*	Removes the port channel and deletes the associated channel group.

Example

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
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Specifies the interface that you want to add to a channel group and enters the interface configuration mode.

	Command or Action	Purpose
Step 3	(Optional) switch(config-if)# **switchport mode trunk**	Configures the interface as a trunk port.
Step 4	(Optional) switch(config-if)# **switchport trunk** {**allowed vlan** *vlan-id* | **native vlan** *vlan-id*}	Configures necessary parameters for a trunk port.
Step 5	switch(config-if)# **channel-group** *channel-number*	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.
Step 6	(Optional) switch(config-if)# **no channel-group**	Removes the port from the channel group. The port reverts to its original configuration.

Example

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-ip-gre** | **destination-mac** | **destination-port** | **source-dest-ip** | **source-dest-ip-gre** | **source-dest-mac** | **source-dest-port** | **source-ip** | **source-ip-gre** | **source-mac** | **source-port**] **symmetric** | **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
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **port-channel load-balance ethernet** {[**destination-ip** | **destination-ip-gre** | **destination-mac** | **destination-port** | **source-dest-ip** | **source-dest-ip-gre** | **source-dest-mac** | **source-dest-port** | **source-ip** | **source-ip-gre** | **source-mac** | **source-port**] **symmetric** | **crc-poly**}	Specifies the load-balancing algorithm and hash for the device. The range depends on the device. The default is **source-dest-mac**. **Note** The optional **destination-ip-gre**, **source-dest-ip-gre** and **source-ip-gre** keywords are used to include the NVGRE key in the hash computation. Inclusion of the NVGRE key is not enabled by default in the case of port channels. You must configure it explicitly by using these optional keywords. The optional **symmetric** keyword is used to enable or disable symmetric hashing. Symmetric hashing forces bi-directional traffic to use the same physical interface. Only the following load-balancing algorithms support symmetric hashing: • source-dest-ip-only • source-dest-port-only • source-dest-ip • source-dest-port • source-dest-ip-gre
Step 3	(Optional) switch(config)# **no port-channel load-balance ethernet**	Restores the default load-balancing algorithm of source-dest-mac.
Step 4	(Optional) switch# **show port-channel load-balance**	Displays the port-channel load-balancing algorithm.

Example

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
```

This example shows how to configure symmetric hashing for port channels:

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

Enabling LACP

LACP is disabled by default; you must enable LACP before you begin LACP configuration. If you have any LACP port channels configured, LACP cannot be disabled.

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
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature lacp**	Enables LACP on the switch.
Step 3	(Optional) switch(config)# **show feature**	Displays enabled features.

Example

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
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Specifies the interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **channel-group** *channel-number* [**force**] [**mode** {**on** | **active** | **passive**}]	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. **force**—Specifies that the LAN port be forcefully added to the channel group. **mode**—Specifies the port channel mode of the interface. **active**—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. **on**—(Default mode) Specifies that all port channels that are not running LACP remain in this mode. **passive**—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. When you run port channels with no associated protocol, the channel mode is always on.
Step 4	switch(config-if)# **no channel-group** *number* **mode**	Returns the port mode to on for the specified interface.

Example

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
```

Configuring LACP Port Channel MinLinks

The MinLink feature works only with LACP port channels. The device allows you to configure this feature in non-LACP port channels, but the feature is not operational.

Important We recommend that you configure the LACP MinLink feature on both ends of your LACP port channel, that is, on both the switches. Configuring the **lacp min-links** command on only one end of the port channel might result in link flapping.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface port-channel** *number*
3. switch(config-if)# [**no**] **lacp min-links** *number*
4. (Optional) switch(config)# **show running-config interface port-channel** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *number*	Specifies the interface to configure.
Step 3	switch(config-if)# [**no**] **lacp min-links** *number*	Configures the number of minimum links. The default value for *number* is 1. The range is from 1 to 32. Use the **no** form of this command to disable this feature.
Step 4	(Optional) switch(config)# **show running-config interface port-channel** *number*	Displays the port channel configuration of the interface.

Example

This example shows how to configure the minimum number of links that must be up for the bundle as a whole to be labeled *up*:

```
switch#configure terminal
switch(config)#interface port-channel 3
switch(config-if)#lacp min-links 3
switch(config)#show running-config interface port-channel 3
```

Configuring the LACP Port-Channel MaxBundle

You can configure the LACP maxbundle feature. Although minimum links and maxbundles work only in LACP, you can enter the CLI commands for these features for non-LACP port channels, but these commands are nonoperational.

Note Use the **no lacp max-bundle** command to restore the default port-channel max-bundle configuration.

Command	Purpose
no lacp max-bundle **Example:** `switch(config)# no lacp max-bundle`	Restores the default port-channel max-bundle configuration.

Before you begin

Ensure that you are in the correct port-channel interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *number*
3. **lacp max-bundle** *number*
4. **show running-config interface port-channel** <*number*>

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface port-channel** *number*
	Example:
	`switch(config)# interface port-channel 3`
	`switch(config-if)#`
		Specifies an interface to configure.

Step 3	**lacp max-bundle** *number* **Example:** `switch(config-if)# lacp max-bundle <number>`	Configures the maximum number of active bundled LACP ports that are allowed in a port channel. The default value for the port-channel max-bundle is 32. The allowed range is from 1 to 32. **Note** Even if the default value is 16, the number of active members in a port channel is the minimum of the *pc_max_links_config* and *pc_max_active_members* that is allowed in the port channel.
Step 4	**show running-config interface port-channel** <*number*> **Example:**	(Optional) Displays the port-channel configuration for the interface.

	Command or Action	Purpose
	`switch(config-if)# show running-config interface port-channel 3`	

Example

This example shows how to configure the maximum number of active bundled LACP ports:

```
switch# configure terminal
switch# interface port-channel 3
switch (config-if)# lacp max-bundle 3
switch (config-if)# show running-config interface port-channel 3
```

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

Example

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
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **lacp system-priority** *priority*	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.
Step 3	(Optional) switch# **show lacp system-identifier**	Displays the LACP system identifier.

Example

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
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot*/*port*	Specifies the interface to configure, and enters the interface configuration mode.
Step 3	switch(config-if)# **lacp port-priority** *priority*	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.

Example

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

By default, LACP graceful convergence is enabled. In situations where you need to support LACP interoperability with devices where the graceful failover defaults may delay the time taken for a disabled port to be brought down or cause traffic from the peer to be lost, you can disable convergence. If the downstream access switch is not a Cisco Nexus device, disable the LACP graceful convergence option.

Note The port channel has to be in the administratively down state before the command can be run.

Before you begin

Enable LACP.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface port-channel** *number*
	Example:
	`switch(config)# interface port-channel 1`
	`switch(config-if)#`
		Specifies the port channel interface to configure and enters the interface configuration mode.

Step 3	**shutdown** **Example:** `switch(config-if) shutdown`	Administratively shuts down the port channel.
Step 4	**no lacp graceful-convergence** **Example:** `switch(config-if)# no lacp graceful-convergence`	Disables LACP graceful convergence on the port channel.
Step 5	**no shutdown** **Example:** `switch(config-if) no shutdown`	Brings up the port channel administratively.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to disable 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
```

Reenabling LACP Graceful Convergence

If the default LACP graceful convergence is once again required, you can reenable convergence.

SUMMARY STEPS

1. **configure terminal**
2. **interface port-channel** *number*

3. **shutdown**
4. **lacp graceful-convergence**
5. **no shutdown**
6. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	**interface port-channel** *number*
	Example:
	`switch(config)# interface port-channel 1`
	`switch(config-if)#`
		Specifies the port channel interface to configure and enters the interface configuration mode.

Step 3	**shutdown** **Example:** `switch(config-if) shutdown`	Administratively shuts down the port channel.
Step 4	**lacp graceful-convergence** **Example:** `switch(config-if)# lacp graceful-convergence`	Enables LACP graceful convergence on the port channel.
Step 5	**no shutdown** **Example:** `switch(config-if) no shutdown`	Brings the port channel administratively up.
Step 6	**copy running-config startup-config** **Example:** `switch(config)# copy running-config startup-config`	(Optional) Copies the running configuration to the startup configuration.

Example

This example shows how to enable 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
```

Verifying Port Channel Configuration

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

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

Triggering the Port Channel Membership Consistency Checker

You can manually trigger the port channel membership consistency checker to compare the hardware and software configuration of all ports in a port channel and display the results. To manually trigger the port channel membership consistency checker and display the results, use the following command in any mode:

SUMMARY STEPS

1. switch# **show consistency-checker membership port-channels**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **show consistency-checker membership port-channels**	Starts a port channel membership consistency check on the member ports of a port channel and displays its results.

Example

This example shows how to trigger a port channel membership consistency check and display its results:

```
switch# show consistency-checker membership port-channels
Checks: Trunk group and trunk membership table.
Consistency Check: PASSED
 No Inconsistencies found for port-channel1111:
  Module:1, Unit:0
      ['Ethernet1/4', 'Ethernet1/5', 'Ethernet1/6']
 No Inconsistencies found for port-channel2211:
  Module:1, Unit:0
      ['Ethernet1/7', 'Ethernet1/8', 'Ethernet1/9', 'Ethernet1/10']
 No Inconsistencies found for port-channel3311:
  Module:1, Unit:0
      ['Ethernet1/11', 'Ethernet1/12', 'Ethernet1/13', 'Ethernet1/14']
 No Inconsistencies found for port-channel4095:
  Module:1, Unit:0
      ['Ethernet1/33', 'Ethernet1/34', 'Ethernet1/35', 'Ethernet1/36', 'Ethernet1
/37', 'Ethernet1/38', 'Ethernet1/39', 'Ethernet1/40', 'Ethernet1/41', 'Ethernet1
/42', 'Ethernet1/43', 'Ethernet1/44', 'Ethernet1/45', 'Ethernet1/46', 'Ethernet1
/47', 'Ethernet1/48', 'Ethernet1/29', 'Ethernet1/30', 'Ethernet1/31', 'Ethernet1
/32']
```

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.

The **show port-channel load-balance** command supports only unicast traffic hashing. Multicast traffic hashing is not supported.

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

Command	Purpose
switch# **show port-channel load-balance forwarding-path interface port-channel** *port-channel-id* **vlan** vlan-id **dst-ip src-ip dst-mac src-mac l4-src-port** *port-id* **l4-dst-port** *port-id* **ether-type** *ether-type* **ip-proto** *ip-proto*	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
l4-src-port 0 l4-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
```

Port Profiles

You can create a port profile that contains many interface commands and apply that port profile to a range of interfaces. Each port profile can be applied only to a specific type of interface; the choices are as follows:

- Ethernet
- VLAN network interface
- Port channel

When you choose Ethernet or port channel as the interface type, the port profile is in the default mode which is Layer 3. Enter the **switchport** command to change the port profile to Layer 2 mode.

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 system applies all the commands in that port profile to the interfaces. Additionally, 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.

The system applies the commands inherited by the interface or range of interfaces according to the following guidelines:

- 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 port-profile command is explicitly overridden by the default command.
- When a range of interfaces inherits a second port profile, the commands of the initial port profile override the commands of the second port profile if there is a conflict.
- 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 you specify.

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 would then enable that port profile for the configurations to take effect on the specified interfaces.

If you inherit one or more port profiles onto an original port profile, only the last inherited port profile must be enabled; the system assumes that the underlying port profiles are enabled.

When you remove a port profile from a range of interfaces, the system undoes the configuration from the interfaces first and then removes the port-profile link itself. Also, when you remove a port profile, the system 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 also 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.

Just as in the device, you can enter a configuration for an object in port profiles without that object being applied to interfaces yet. For example, you can configure a virtual routing and forward (VRF) instance without it being applied to the system. If you then delete that VRF and related configurations from the port profile, the system is unaffected.

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

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

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

Configuring Port Profiles

You can apply several configuration parameters to a range of interfaces simultaneously. All the interfaces in the range must be the same type. You can also inherit the configurations from one port profile into another port profile. The system supports four levels of inheritance.

Creating a Port Profile

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

Note Port profile names can include only the following characters:

- a-z
- A-Z
- 0-9
- No special characters are allowed, except for the following:
    - .
    - -
    - _

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	**configure terminal**	Enters the global configuration mode.
Step 2	**port-profile** [**type** {**ethernet** | **interface-vlan** | **port-channel**}] *name*	Creates and names a port profile for the specified type of interface and enters the port-profile configuration mode.
Step 3	**exit**	Exits the port-profile configuration mode.
Step 4	(Optional) **show port-profile**	Displays the port-profile configuration.
Step 5	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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-ppm)#
```

Entering Port-Profile Configuration Mode and Modifying a Port Profile

You can enter the port-profile configuration mode and modify a port profile. To modify the port profile, you must be in the port-profile configuration mode.

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	**configure terminal**	Enters the global configuration mode.
Step 2	**port-profile** [**type** {**ethernet** | **interface-vlan** | **port-channel**}] *name*	Enters the port-profile configuration mode for the specified port profile and allows you to add or remove configurations to the profile.
Step 3	**exit**	Exits the port-profile configuration mode.
Step 4	(Optional) **show port-profile**	Displays the port-profile configuration.
Step 5	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to enter the port-profile configuration mode for the specified port profile and bring all the interfaces administratively up:

```
switch# configure terminal
switch(config)# port-profile type ethernet test
switch(config-ppm)# no shutdown
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 the interfaces must be the same type.

SUMMARY STEPS

1. **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**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters the global configuration mode.
Step 2	**interface** [**ethernet** *slot/port* | **interface-vlan** *vlan-id* | **port-channel** *number*]	Selects the range of interfaces.
Step 3	**inherit port-profile** *name*	Assigns the specified port profile to the selected interfaces.
Step 4	**exit**	Exits the port-profile configuration mode.
Step 5	(Optional) **show port-profile**	Displays the port-profile configuration.
Step 6	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to assign the port profile named adam to Ethernet interfaces 7/3 to 7/5, 10/2, and 11/20 to 11/25:

```
switch# configure terminal
switch(config)# interface ethernet7/3-5, ethernet10/2, ethernet11/20-25
switch(config-if)# inherit port-profile adam
switch(config-if)#
```

Enabling a Specific Port Profile

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 before you enable that port profile. You would then enable that port profile for the configurations to take effect on the specified interfaces.

If you inherit one or more port profiles onto an original port profile, only the last inherited port profile must be enabled; the system assumes that the underlying port profiles are enabled.

You must be in the port-profile configuration mode to enable or disable port profiles.

SUMMARY STEPS

1. **configure terminal**

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters the global configuration mode.
Step 2	**port-profile** [**type** {**ethernet** | **interface-vlan** | **port-channel**}] *name*	Creates and names a port profile for the specified type of interface and enters the port-profile configuration mode.
Step 3	**state enabled**	Enables that port profile.
Step 4	**exit**	Exits the port-profile configuration mode.
Step 5	(Optional) **show port-profile**	Displays the port-profile configuration.
Step 6	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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

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

Inheriting a Port Profile

You can inherit a port profile onto an existing port profile. The system 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
Step 1	**configure terminal**	Enters the global configuration mode.
Step 2	**port-profile** *name*	Enters the port-profile configuration mode for the specified port profile.
Step 3	**inherit port-profile** *name*	Inherits another port profile onto the existing one. The original port profile assumes all the configurations of the inherited port profile.
Step 4	**exit**	Exits the port-profile configuration mode.
Step 5	(Optional) **show port-profile**	Displays the port-profile configuration.
Step 6	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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)#
```

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. You do this configuration in the interfaces configuration mode.

SUMMARY STEPS

1. **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) **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure terminal**	Enters the global configuration mode.

	Command or Action	Purpose
Step 2	**interface** [**ethernet** *slot/port* | **interface-vlan** *vlan-id* | **port-channel** *number*]	Selects the range of interfaces.
Step 3	**no inherit port-profile** *name*	Un-assigns the specified port profile to the selected interfaces.
Step 4	**exit**	Exits the port-profile configuration mode.
Step 5	(Optional) **show port-profile**	Displays the port-profile configuration.
Step 6	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

This example shows how to unassign the port profile named adam to Ethernet interfaces 7/3 to 7/5, 10/2, and 11/20 to 11/25:

```
switch# configure terminal
switch(config)# interface ethernet 7/3-5, 10/2, 11/20-25
switch(config-if)# no inherit port-profile adam
switch(config-if)#
```

Removing an Inherited Port Profile

You can remove an inherited port profile. You do this configuration in the port-profile mode.

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	**configure terminal**	Enters the global configuration mode.
Step 2	**port-profile** *name*	Enters the port-profile configuration mode for the specified port profile.
Step 3	**no inherit port-profile** *name*	Removes an inherited port profile from this port profile.
Step 4	**exit**	Exits the port-profile configuration mode.

	Command or Action	Purpose
Step 5	(Optional) **show port-profile**	Displays the port-profile configuration.
Step 6	(Optional) **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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)#
```

CHAPTER 6

Configuring Virtual Port Channels

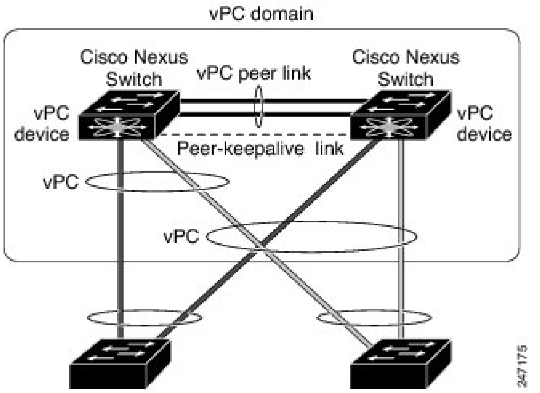
- Information About vPCs, on page 92
- Per-VLAN Consistency Check, on page 97
- vPC Auto-Recovery, on page 97
- vPC Peer Links, on page 97
- vPC Number, on page 99
- vPC Interactions with Other Features, on page 99
- vPC Forklift Upgrade Scenario, on page 100
- Guidelines and Limitations for vPCs, on page 103
- Verifying the vPC Configuration, on page 104
- Viewing the Graceful Type-1 Check Status, on page 104
- Viewing a Global Type-1 Inconsistency, on page 105
- Viewing an Interface-Specific Type-1 Inconsistency, on page 106
- Viewing a Per-VLAN Consistency Status, on page 107
- vPC Default Settings, on page 110
- Configuring vPCs, on page 110
- Configuring a vPC Keepalive Link and Messages, on page 112
- Creating a vPC Peer Link, on page 115
- Checking the Configuration Compatibility, on page 116
- Enabling vPC Auto-Recovery, on page 117
- Configuring the Restore Time Delay, on page 118
- Excluding VLAN Interfaces from Shutting Down a vPC Peer Link Fails, on page 119
- Configuring the VRF Name, on page 119
- Moving Other Port Channels into a vPC, on page 120
- Manually Configuring a vPC Domain MAC Address, on page 121
- Manually Configuring the System Priority, on page 122
- Manually Configuring a vPC Peer Switch Role, on page 123
- Configuring Layer 3 over vPC, on page 124

Information About vPCs

vPC Overview

A virtual port channel (vPC) allows links that are physically connected to two Cisco Nexus devices 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. 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 5: 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 32 active links in a single 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.
- 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.

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.

If one of the vPC peer switches fails, the vPC peer switch on the other side of the vPC peer link senses the failure when it does not receive any peer-keepalive messages. The default interval time for the vPC peer-keepalive message is 1 second. You can configure the interval between 400 milliseconds and 10 seconds. You can also configure a timeout value with a range of 3 to 20 seconds; the default timeout value is 5 seconds. The peer-keepalive status is checked only when the peer-link goes down.

The vPC peer-keepalive can be carried either in the management or default VRF on the Cisco Nexus device. When you configure the switches to use the management VRF, the source and destination for the keepalive messages are the mgmt 0 interface IP addresses. When you configure the switches to use the default VRF, an SVI must be created to act as the source and destination addresses for the vPC peer-keepalive messages. Ensure that both the source and destination IP addresses used for the peer-keepalive messages are unique in your network and these IP addresses are reachable from the VRF associated with the vPC peer-keepalive link.

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.

New Type 2 Consistency Check on the vPC Port-Channels

A new type 2 consistency check has been added to validate the switchport mac learn settings on the vPC port-channels. The CLI **show vpc consistency-check vPC <vpc no.>** has been enhanced to display the local and peer values of the switchport mac-learn configuration. Because it is a type 2 check, vPC is operationally up even if there is a mismatch between the local and the peer values, but the mismatch can be displayed from the CLI output.

```
switch# sh vpc consistency-parameters vpc 1112

    Legend:
        Type 1 : vPC will be suspended in case of mismatch

Name                                Type            Local Value                 Peer Value
-------------                        ----            ----------------------
-----------------------
Shut Lan                             1               No                         No
STP Port Type                        1               Default                     Default
STP Port Guard                       1               None                        None
STP MST Simulate PVST                1               Default                     Default
nve configuration                    1               nve                        nve
lag-id                               1               [(fa0,                      [(fa0,
                                                     0-23-4-ee-be-64, 8458,
0-23-4-ee-be-64, 8458,
                                                     0, 0), (8000,              0, 0),
(8000,
                                                     f4-4e-5-84-5e-3c, 457,
f4-4e-5-84-5e-3c, 457,
                                                     0, 0)]                     0, 0)]
mode                                 1               active                      active
Speed                                1               10 Gb/s                    10 Gb/s
Duplex                               1               full                        full
Port Mode                            1               trunk                       trunk
Native Vlan                          1               1                          1
MTU                                  1               1500                        1500
Admin port mode                      1
Switchport MAC Learn                 2               Enable                     Disable>
Newly added consistency parameter
vPC card type                        1               Empty                      Empty
Allowed VLANs                        -               311-400                     311-400
Local suspended VLANs                -               -
```

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

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.
- 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.

Per-VLAN Consistency Check

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

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.

The vPC auto-recovery feature is enabled by default.

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 reenable 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

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.

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.

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 Forklift Upgrade Scenario

The following describes a scenario for migrating from a pair of Cisco Nexus 3600 platform switches in a vPC topology to a different pair of Cisco Nexus 3600 platform switches.

Considerations for a vPC forklift upgrade:

- vPC Role Election and Sticky-bit

When the two vPC systems are joined to form a vPC domain, priority decides which device is the vPC primary and which is the vPC secondary. When the primary device is reloaded, the system comes back online and connectivity to the vPC secondary device (now the operational primary) is restored. The operational role of the secondary device (operational primary) does not change (to avoid unnecessary disruptions). This behavior is achieved with a sticky-bit, where the sticky information is not saved in the startup configuration. This method makes the device that is up and running win over the reloaded device. Hence, the vPC primary becomes the vPC operational secondary. Sticky-bit is also set when a vPC node comes up with peer-link and peer-keepalive down and it becomes primary after the auto recovery period.

- vPC Delay Restore

  The delay restore timer is used to delay the vPC from coming up on the restored vPC peer device after a reload when the peer adjacency is already established.

  To delay the VLAN interfaces on the restored vPC peer device from coming up, use the **interfaces-vlan** option of the **delay restore** command.

- vPC Auto-Recovery

  During a data center power outage when both vPC peer switches go down, if only one switch is restored, the auto-recovery feature allows that switch to assume the role of the primary switch and the vPC links come up after the auto-recovery time period. The default auto-recovery period is 240 seconds.

The following example is a migration scenario that replaces vPC peer nodes Node1 and Node2 with New_Node1 and New_Node2.

	Migration Step	**Expected Behavior**	**Node1 Configured role (Ex: role priority 100)**	**Node1 Operational role**	**Node2 Configured role (Ex: role priority 200)**	**Node2 Operational role**
1	Initial state	Traffic is forwarded by both vPC peers – Node1 and Node2. Node1 is primary and Node2 is secondary.	primary	Primary Sticky bit: False	secondary	Secondary Sticky bit: False
2	Node2 replacement – Shut all vPCs and uplinks on Node2. Peer-link and vPC peer-keepalive are in administrative up state.	Traffic converged on Primary vPC peer Node1.	primary	Primary Sticky bit: False	secondary	Secondary Sticky bit: False
3	Remove Node2.	Node1 will continue to forward traffic.	primary	Primary Sticky bit: False	n/a	n/a

	Migration Step	Expected Behavior	Node1 Configured role (Ex: role priority 100)	Node1 Operational role	Node2 Configured role (Ex: role priority 200)	Node2 Operational role
4	Configure New_Node2. Copy the configuration to startup config. vPC peer-link and peer-keepalive in administrative up state. Power off New_Node2. Make all connections. Power on New_Node2.	New_Node2 will come up as secondary. Node1 continue to be primary. Traffic will continue to be forwarded on Node01.	primary	Primary Sticky bit: False	secondary	Secondary Sticky bit: False
5	Bring up all vPCs and uplink ports on New_Node2.	Traffic will be forwarded by both Node 1 and New_Node2.	primary	Primary Sticky bit: False	secondary	Secondary Sticky bit: False
6	Node1 replacement - Shut vPCs and uplinks on Node1.	Traffic will converge on New_Node2.	primary	Primary Sticky bit: False	secondary	Secondary Sticky bit: False
7	Remove Node1.	New_Node2 will become secondary, operational primary and sticky bit will be set to True.	n/a	n/a	secondary	Primary Sticky bit: True
8	Configure New_Node1. Copy running to startup. Power off the new Node1. Make all connections. Power on New_Node1.	New_Node1 will come up as primary, operational secondary.	primary	Secondary Sticky bit: False	secondary	Primary Sticky bit: True
9	Bring up all vPCs and uplink ports on New_Node1.	Traffic will be forwarded by both New Node1 and new Node2.	primary	Secondary Sticky bit: False	secondary	Primary Sticky bit: True

Note If you prefer to have the configured secondary node as the operational secondary and the configured primary as the operational primary, then Node2 can be reloaded at the end of the migration. This is optional and does not have any functional impact.

Guidelines and Limitations for vPCs

vPCs have the following configuration guidelines and limitations:

- vPC is not supported between different types of Cisco Nexus 3000 Series switches.
- VPC peers should have same reserved VLANs for VXLAN. Different reserved VLANs on the peers may lead to undesired behavior with VXLAN.
- The output of the **sh vpc brief** CLI command displays two additional fields, Delay-restore status and Delay-restore SVI status.
- 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.
- 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.
- Only port channels can be in vPCs. A vPC can be configured on a normal port channel (switch-to-switch vPC topology) and on a port channel host interface (host interface vPC topology).
- 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 might experience minimal traffic disruption while configuring vPCs.
- You should configure all port channels in the vPC using LACP with the interfaces in active mode.
- You might experience traffic disruption when the first member of a vPC is brought up.
- OSPF over vPC and BFD with OSPF are supported on Cisco Nexus 3000 Series switches.

  SVI limitation: When a BFD session is over SVI using virtual port-channel(vPC) peer-link, the BFD echo function is not supported. You must disable the BFD echo function for all sessions over SVI between vPC peer nodes using **no bfd echo** at the SVI configuration level.
- When a Layer 3 link is used for peer-keepalive instead of the mgmt interface, and the CPU queues are congested with control plane traffic, vPC peer-keepalive packets could be dropped. The CPU traffic includes routing protocol, ARP, Glean, and IPMC miss packets. When the peer-keepalive interface is a Layer 3 link instead of a mgmt interface, the vPC peer-keepalive packets are sent to the CPU on a low-priority queue.

  If a Layer 3 link is used for vPC peer-keepalives, configure the following ACL to prioritize the vPC peer-keepalive:

```
ip access-list copp-system-acl-routingproto2
30 permit udp any any eq 3200
```

Here, 3200 is the default UDP port for keepalive packets. This ACL must match the configured UDP port in case the default port is changed.

Verifying the vPC Configuration

Use the following commands to display vPC configuration information:

Command	Purpose
switch# **show feature**	Displays whether vPC is enabled or not.
switch# **show port-channel capacity**	Displays how many EtherChannels are configured and how many are still available on the switch.
switch# **show running-config vpc**	Displays running configuration information for vPCs.
switch# **show vpc brief**	Displays brief information on the vPCs.
switch# **show vpc consistency-parameters**	Displays the status of those parameters that must be consistent across all vPC interfaces.
switch# **show vpc peer-keepalive**	Displays information on the peer-keepalive messages.
switch# **show vpc role**	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# **show vpc statistics**	Displays statistics on the vPCs. **Note** 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
Auto-recovery status              : Disabled
Delay-restore status              : Timer is off.(timeout = 30s)
Delay-restore SVI status          : Timer is off.(timeout = 10s)

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
Auto-recovery status              : Disabled
Delay-restore status              : Timer is off.(timeout = 30s)
Delay-restore SVI status          : Timer is off.(timeout = 10s)

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
Auto-recovery status                : Disabled
Delay-restore status                : Timer is off.(timeout = 30s)
Delay-restore SVI status            : Timer is off.(timeout = 10s)

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.

Example

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
Auto-recovery status              : Disabled
Delay-restore status              : Timer is off.(timeout = 30s)
Delay-restore SVI status          : Timer is off.(timeout = 10s)

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                    : 1
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 and BDs : -
Graceful Consistency Check       : Enabled
Auto-recovery status             : Enabled, timer is off.(timeout = 240s)
Delay-restore status             : Timer is off.(timeout = 30s)
Delay-restore SVI status         : Timer is off.(timeout = 10s)

vPC Peer-link status
---------------------------------------------------------------------
id   Port   Status Active vlans
--   ----   ------ --------------------------------------------------
1    Po1000 up     1-5,8,11-19


vPC status
------------------------------------------------------
id    Port         Status Consistency Active VLANs
----- ------------ ------ ----------- ----------------
```

```
101   Po101        up     success     1-5,8,11-19
102   Po102        up     success     1-5,8,11-19
```

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
Auto-recovery status                : Disabled
Delay-restore status                : Timer is off.(timeout = 30s)
Delay-restore SVI status            : Timer is off.(timeout = 10s)

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 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 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
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature vpc**	Enables vPCs on the switch.
Step 3	(Optional) switch# **show feature**	Displays which features are enabled on the switch.
Step 4	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no feature vpc**	Disables vPCs on the switch.
Step 3	(Optional) switch# **show feature**	Displays which features are enabled on the switch.
Step 4	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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

Example

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.

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
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	Creates a vPC domain on the switch if it does not already exist, and enters the vpc-domain configuration mode.
Step 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**}]	Configures the IPv4 address for the remote end of the vPC peer-keepalive link. **Note** 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.
Step 4	(Optional) switch(config-vpc-domain)# **vpc peer-keepalive destination** *ipaddress* **source** *ipaddress*	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.
Step 5	(Optional) switch# **show vpc peer-keepalive**	Displays information about the configuration for the keepalive messages.
Step 6	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface port-channel** *channel-number*	Selects the EtherChannel that you want to use as the vPC peer link for this switch, and enters the interface configuration mode.
Step 3	switch(config-if)# **vpc peer-link**	Configures the selected EtherChannel as the vPC peer link, and enters the vpc-domain configuration mode.
Step 4	(Optional) switch# **show vpc brief**	Displays information about each vPC, including information about the vPC peer link.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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.

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.

SUMMARY STEPS

1. switch# **show vpc consistency-parameters**{**global**|**interface port-channel***channel-number*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **show vpc consistency-parameters**{**global**|**interface port-channel***channel-number*}	Displays the status of those parameters that must be consistent across all vPC interfaces.

Example

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     2     (F, F, F, F, F, F)     (100, 0, 0, 0, 0, 0)
Priority)
Output Queuing (Bandwidth)  2     (100, 0, 0, 0, 0, 0)   (100, 0, 0, 0, 0, 0)
Output Queuing (Absolute    2     (F, F, F, F, F, F)     (100, 0, 0, 0, 0, 0)
Priority)
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  1
  VLAN Mapping

STP Loopguard               1     Disabled               Disabled
STP Bridge Assurance        1     Enabled                Enabled
STP Port Type, Edge         1     Normal, Disabled,      Normal, Disabled,
BPDUFilter, Edge BPDUGuard        Disabled               Disabled
STP MST Simulate PVST       1     Enabled                Enabled
Allowed VLANs               -     1,624                  1
Local suspended VLANs       -     624                    -
switch#
```

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
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	Enters vpc-domain configuration mode for an existing vPC domain.
Step 3	switch(config-vpc-domain)# **auto-recovery reload-delay** *delay*	Enables the auto-recovery feature and sets the reload delay period. The default is disabled.

Example

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


feature vpc
vpc domain 10
```

```
peer-keepalive destination 10.193.51.170
auto-recovery
```

Configuring the Restore Time Delay

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
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	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)# **delay restore** *time*	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.
Step 4	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	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))# **dual-active exclude interface-vlan** *range*	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.

Example

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# **ping** *ipaddress* **vrf** *vrf-name*	Specifies the virtual routing and forwarding (VRF) name to use. The VRF name is case sensitive and can be a maximum of 32 characters..

Example

This example shows how to specifiy 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
```

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
Step 1	switch# **configure terminal**	Enters global configuration mode.

	Command or Action	Purpose
Step 2	switch(config)# **interface port-channel** *channel-number*	Selects the port channel that you want to put into the vPC to connect to the downstream switch, and enters interface configuration mode. **Note** A vPC can be configured on a normal port channel (physical vPC topology) and on a port channel host interface (host interface vPC topology)
Step 3	switch(config-if)# **vpc** *number*	Configures the selected port channel into the vPC to connect to the downstream switch. The range is from 1 to 4096. The vPC *number* 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.
Step 4	(Optional) switch# **show vpc brief**	Displays information about each vPC.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	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 *domain-id*; the range is from 1 to 1000.
Step 3	switch(config-vpc-domain)# **system-mac** *mac-address*	Enters the MAC address that you want for the specified vPC domain in the following format: aaaa.bbbb.cccc.
Step 4	(Optional) switch# **show vpc role**	Displays the vPC system MAC address.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	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 *domain-id*; the range is from 1 to 1000.
Step 3	switch(config-vpc-domain)# **system-priority** *priority*	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.
Step 4	(Optional) switch# **show vpc brief**	Displays information about each vPC, including information about the vPC peer link.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vpc domain** *domain-id*	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 *domain-id*; the range is from 1 to 1000.
Step 3	switch(config-vpc-domain)# **role priority** *priority*	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.
Step 4	(Optional) switch# **show vpc brief**	Displays information about each vPC, including information about the vPC peer link.
Step 5	(Optional) switch# **copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

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 Layer 3 over vPC

Before you begin

Ensure that the peer-gateway feature is enabled and it is configured on both the peers and both the peers run an image that supports Layer 3 over vPC. If you enter the **layer3 peer-router** command without enabling the peer-gateway feature, a syslog message is displayed recommending you to enable the peer-gateway feature.

Ensure that the peer link is up.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vpc domain** *domain-id*
3. switch(config-vpc-domain)#**layer3 peer-router**

4. switch(config-vpc-domain)# **exit**
5. (Optional) switch#**show vpc brief**
6. (Optional) switch#**copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

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

Step 2	switch(config)# **vpc domain** *domain-id*
	Example:
	`switch(config)# vpc domain 5`
	`switch(config-vpc-domain)#`
		Creates a vPC domain if it does not already exist, and enters the vpc-domain configuration mode. There is no default; the range is from <1 to 1000>.

Step 3	switch(config-vpc-domain)#**layer3 peer-router**	Enables the Layer 3 device to form peering adjacency with both the peers. **Note** Configure this command in both the peers. If you configure this command only on one of the peers or you disable it on one peer, the operational state of layer 3 peer-router gets disabled. You get a notification when there is a change in the operational state.
Step 4	switch(config-vpc-domain)# **exit**	Exits the vpc-domain configuration mode.
Step 5	(Optional) switch#**show vpc brief**	Displays brief information about each vPC domain.
Step 6	(Optional) switch#**copy running-config startup-config**	Copies the running configuration to the startup configuration.

Example

The following example shows how to configure Layer 3 over vPC feature:

```
switch# configure terminal
switch(config)# vpc domain 5
switch(config-vpc-domain)# layer3 peer-router

switch(config-vpc-domain)# exit

switch(config)#
```

This example shows how to verify if the Layer 3 over vPC feature is configured. The **Operational Layer3 Peer** is enabled or disabled depending up on how the operational state of Layer 3 over vPC is configured.

```
switch# show vpc brief
```

```
vPC domain id : 5
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 : Enabled
Peer gateway excluded VLANs : -
Dual-active excluded VLANs : -
Graceful Consistency Check : Enabled
Auto-recovery status : Enabled (timeout = 240 seconds)
Operational Layer3 Peer : Enabled
```

INDEX

A

- adding ports 67
  - port channels 67

B

- bandwidth 42
  - configuring 42

C

- channel mode 70
  - port channels 70
- channel modes 64
  - port channels 64
- configuration 52
  - Layer 3 interfaces 52
    - verifying 52
- configuration examples 55
  - Layer 3 interfaces 55
- configuring 25, 28, 40–43, 45, 74–75
  - description parameter 28
  - error-disabled recovery interval 25
  - interface bandwidth 42
  - LACP fast timer rate 74
  - LACP port priority 75
  - loopback interfaces 45
  - routed interfaces 40
  - subinterfaces 41
  - VLAN interfaces 43
- Configuring a DHCP client on an interface 52
- configuring LACP 69

D

- debounce timer 11
  - parameters 11
- debounce timer, configuring 27
  - Ethernet interfaces 27
- default interface 11
- default settings 40
  - Layer 3 interfaces 40
- delay restore 101
- disabling 18, 23, 26, 28, 111
  - CDP 23
  - error-disabled recovery 26
  - ethernet interfaces 28
  - link negotiation 18
  - vPCs 111
- downlink delay 12

E

- enabling 23–24
  - CDP 23
  - error-disabled detection 23
  - error-disabled recovery 24
- Ethernet interfaces 8, 27
  - debounce timer, configuring 27
  - interface speed 8

I

- interface ethernet 30–31
- interface information, displaying 32
  - layer 2 32
- interface MAC address, configuring 47
- interface port-channel 73, 76–78
- interface speed 8, 15
  - configuring 15
  - Ethernet interfaces 8
- interfaces 5–6, 35, 37, 39, 42–43, 45–46, 53, 55
  - assigning to a VRF 46
  - chassis ID 5
  - configuring bandwidth 42
  - Layer 3 35, 53, 55
    - configuration examples 55
    - monitoring 53
  - loopback 39, 45
  - options 5
  - routed 35
  - tunnel 39
  - UDLD 6
  - VLAN 37, 43
    - configuring 43
- interfaces-vlan 101

L

LACP 58, 62–63, 65, 69, 71
- configuring 69
- marker responders 65
- port channel, minlinks 65, 71
- port channels 62
- system ID 63

LACP fast timer rate 74
- configuring 74

lacp graceful-convergence 78
lacp max-bundle 73
LACP port priority 75
- configuring 75

LACP-enabled vs static 65
- port channels 65

layer 2 9, 20, 32
- interface information, displaying 32
- svi autostate 9
- svi autostate, disabling 20

Layer 3 interfaces 35, 40, 52–53, 55
- configuration examples 55
- configuring routed interfaces 40
- default settings 40
- interfaces 55
  - Layer 3 55
    - related documents 55
- monitoring 53
- related documents 55
- verifying 52

Link Aggregation Control Protocol 58
load balancing 68
- port channels 68
  - configuring 68

loopback interfaces 39, 45
- configuring 45

M

monitoring 53
- Layer 3 interfaces 53

P

parameters, about 11
- debounce timer 11

physical Ethernet settings 13
port channel 79
- verifying configuration 79

port channel, minlinks 65, 71
- LACP 65, 71

port channeling 58
port channels 42, 57–58, 60, 62, 65–68, 70, 120
- adding ports 67
- channel mode 70

port channels *(continued)*
- compatibility requirements 58
- configuring bandwidth 42
- creating 66
- LACP 62
- LACP-enabled vs static 65
- load balancing 60, 68
  - port channels 60
- moving into a vPC 120
- STP 57

R

related documents 55
- Layer 3 interfaces 55

restarting 28
- ethernet interfaces 28

routed interfaces 35, 40, 42
- configuring 40
- configuring bandwidth 42

S

SFP+ transceiver 8
show running-config interface port-channel 73
shutdown 76–78
Small form-factor pluggable (plus) transceiver 8
STP 57
- port channel 57

subinterfaces 36, 41–42
- configuring 41
- configuring bandwidth 42

svi autostate 9
- layer 2 9

SVI autostate disable 40
SVI autostate disable, configuring 51
svi autostate, disabling 20
- layer 2 20

switchport 30–31
switchport dot1q ethertype 30–31
switchport mode 30–31
symmetric hashing 62

T

tunnel interfaces 39

U

UDLD 6–7
- aggressive mode 7
- defined 6
- nonaggressive mode 7

UDLD modeA 13
- configuring 13

Unidirectional Link Detection 6

V

verifying 52
  Layer 3 interface configuration 52
VLAN 37
  interfaces 37
VLAN interfaces 43
  configuring 43
vPC terminology 93
vPCs 120
  moving port channels into 120
VRF 46
  assigning an interface to 46