



Interface and Hardware Components Configuration Guide, Cisco IOS XE Dublin 17.11.x (Catalyst 9200 Switches)

First Published: 2023-03-31

Americas Headquarters

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

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

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

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

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

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

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

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

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

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2023 Cisco Systems, Inc. All rights reserved.



CONTENTS

CHAPTER 1

Configuring Interface Characteristics 1

Information About Interface Characteristics 1

Interface Types 1

Port-Based VLANs 1

Switch Ports 2

Using the Switch USB Ports 5

USB Mini-Type B Console Port 5

Console Port Change Logs 5

USB Type A Port 6

Disabling USB Ports 6

Interface Connections 6

Interface Configuration Mode 7

Default Ethernet Interface Configuration 8

Interface Speed and Duplex Mode 9

Speed and Duplex Configuration Guidelines 9

IEEE 802.3x Flow Control 10

Layer 3 Interfaces 11

How to Configure Interface Characteristics 12

Configuring an Interface 12

Adding a Description for an Interface 13

Configuring a Range of Interfaces 13

Configuring and Using Interface Range Macros 15

Setting the Interface Speed and Duplex Parameters 16

Configuring the IEEE 802.3x Flow Control 17

Configuring a Layer 3 Interface 18

Configuring a Logical Layer 3 GRE Tunnel Interface 19

Configuring SVI Autostate Exclude	21
Shutting Down and Restarting an Interface	22
Configuring the Console Media Type	23
Configuring USB Inactivity Timeout	24
Disabling USB Ports	24
Monitoring Interface Characteristics	25
Monitoring Interface Status	25
Clearing and Resetting Interfaces and Counters	26
Configuration Examples for Interface Characteristics	27
Example: Adding a Description to an Interface	27
Example: Configuring Interfaces on a Stack-Capable Switch	27
Example: Configuring a Range of Interfaces	27
Example: Configuring and Using Interface Range Macros	28
Example: Setting Interface Speed and Duplex Mode	28
Example: Configuring a Layer 3 Interface	28
Example: Configuring the Console Media Type	29
Example: Configuring USB Inactivity Timeout	29
Additional References for Configuring Interface Characteristics	30
Feature History for Configuring Interface Characteristics	30

CHAPTER 2

Configuring Auto-MDIX	31
Prerequisites for Auto-MDIX	31
Restrictions for Auto-MDIX	31
Information About Configuring Auto-MDIX	31
Auto-MDIX on an Interface	32
How to Configure Auto-MDIX	32
Configuring Auto-MDIX on an Interface	32
Example for Configuring Auto-MDIX	33
Auto-MDIX and Operational State	33
Additional References for Auto-MDIX	34
Feature History for Auto-MDIX	34

CHAPTER 3

Configuring Ethernet Management Port	35
Prerequisites for Ethernet Management Port	35

Information About the Ethernet Management Port	35
Ethernet Management Port Direct Connection to a Device	35
Ethernet Management Port Connection to Stack Devices using a Hub	36
Ethernet Management Port and Routing	36
Supported Features on the Ethernet Management Port	36
How to Configure the Ethernet Management Port	37
Disabling and Enabling the Ethernet Management Port	37
Example for Configuring IP Address on Ethernet Management Interface	38
Additional References for Ethernet Management Port	39
Feature History for Ethernet Management Port	39

CHAPTER 4**Checking Port Status and Connectivity 41**

Check Cable Status Using Time Domain Reflectometer	41
Running the TDR Test	41
TDR Guidelines	41
Feature History for Checking Port Status and Connectivity	42

CHAPTER 5**Configuring LLDP, LLDP-MED, and Wired Location Service 43**

Restrictions for LLDP	43
Information About LLDP, LLDP-MED, and Wired Location Service	43
LLDP	43
LLDP Supported TLVs	44
LLDP-MED	44
LLDP-MED Supported TLVs	44
Wired Location Service	46
Default LLDP Configuration	47
How to Configure LLDP, LLDP-MED, and Wired Location Service	47
Enabling LLDP	47
Configuring LLDP Characteristics	48
Configuring LLDP-MED TLVs	50
Configuring Network-Policy TLV	51
Configuring Location TLV and Wired Location Service	52
Enabling Wired Location Service on the Device	54
Configuration Examples for LLDP, LLDP-MED, and Wired Location Service	55

Examples: Configuring Network-Policy TLV	55
Monitoring and Maintaining LLDP, LLDP-MED, and Wired Location Service	56
Additional References for LLDP, LLDP-MED, and Wired Location Service	57
Feature History for LLDP, LLDP-MED, and Wired Location Service	57

CHAPTER 6**Configuring System MTU 59**

Restrictions for System MTU	59
Information About the MTU	59
System MTU Value Application	59
How to Configure MTU	59
Configuring the System MTU	60
Configuring Protocol-Specific MTU	60
Configuration Examples for System MTU	61
Example: Configuring Protocol-Specific MTU	61
Example: Configuring the System MTU	61
Additional References for System MTU	62
Feature History for System MTU	62

CHAPTER 7**Configuring Per-Port MTU 63**

Restrictions for Per-Port MTU	63
Information About Per-Port MTU	63
Configuring Per-Port MTU	64
Example: Configuring Per-Port MTU	64
Example: Verifying Per-Port MTU	65
Example: Disabling Per-Port MTU	65
Feature History for Per-Port MTU	65

CHAPTER 8**Configuring Internal Power Supplies 67**

Information About Internal Power Supplies	67
How to Configure Internal Power Supplies	67
Configuring Internal Power Supply	67
Monitoring Internal Power Supplies	68
Configuration Examples for Internal Power Supplies	68
Additional References for Internal Power Supplies	69

Feature History for Internal Power Supplies 69

CHAPTER 9

Configuring EEE 71

Restrictions for EEE 71

Information About EEE 71

EEE Overview 71

Default EEE Configuration 71

How to Configure EEE 72

Enabling or Disabling EEE 72

Monitoring EEE 73

Configuration Examples for Configuring EEE 73

Additional References for EEE 74

Feature History for Configuring EEE 74

CHAPTER 10

Configuring Power over Ethernet 75

Information About Power over Ethernet 75

PoE and PoE+ Ports 75

Supported Protocols and Standards 75

Powered-Device Detection and Initial Power Allocation 76

Power Management Modes 77

How to Configure PoE and UPOE 79

Configuring a Power Management Mode on a PoE Port 79

Configuring Power Policing 81

Monitoring Power Status 82

Additional References for Power over Ethernet 83

Feature History for Power over Ethernet 83

CHAPTER 11

Configuring Perpetual PoE and Fast POE 85

Restrictions for Perpetual and Fast PoE 85

Information About Perpetual PoE 85

Fast POE 86

Configuring Perpetual and Fast PoE 86

Example: Configuring Perpetual and Fast PoE 87

Feature History for Perpetual PoE and Fast PoE 87

CHAPTER 12	Configuring 2-event Classification	89
	Restrictions for 2-Event Classification	89
	Information about 2-event Classification	89
	Configuring 2-Event Classification	89
	Example: Configuring 2-Event Classification	90
	Feature History for 2-Event Classification	90

CHAPTER 13	Configuring Auto SmartPorts	93
	Restrictions for Auto SmartPorts	93
	Information about Auto SmartPorts	93
	Auto SmartPort Macros	94
	Commands run by CISCO_LIGHT_AUTO_SMARTPORT	94
	Enabling Auto SmartPort	95
	How to Configure Auto SmartPorts	96
	Configuring Mapping Between Event Triggers and Built-in Macros	96
	Configuration Examples for Auto SmartPorts	97
	Example: Enabling Auto SmartPorts	97
	Example: Configuring Mapping Between Event Triggers and Built-In Macros	97
	Feature History for Auto SmartPorts	98

CHAPTER 14	Configuring COAP Proxy Server	99
	Restrictions for the COAP Proxy Server	99
	Information About the COAP Proxy Server	99
	How to Configure the COAP Proxy Server	100
	Configuring the COAP Proxy	100
	Configuring COAP Endpoints	102
	Configuration Examples for the COAP Proxy Server	103
	Examples: Configuring the COAP Proxy Server	103
	Monitoring COAP Proxy Server	107
	Feature History for COAP	108

CHAPTER 15	Configuring an External USB Bluetooth Dongle	109
	Restrictions for Configuring an External USB Bluetooth Dongle	109

Information About External USB Bluetooth Dongle	109
Supported External USB Bluetooth Dongle	109
How to Configure an External USB Bluetooth Dongle on a Switch	110
Verifying Bluetooth Settings on a Device	111
Feature History for Configuring an External Bluetooth Dongle	111

CHAPTER 16**Troubleshooting Interface and Hardware Components 113**

Overview	113
Support Articles	113
Feedback Request	114
Disclaimer and Caution	114



CHAPTER 1

Configuring Interface Characteristics

- [Information About Interface Characteristics, on page 1](#)
- [How to Configure Interface Characteristics, on page 12](#)
- [Configuration Examples for Interface Characteristics, on page 27](#)
- [Additional References for Configuring Interface Characteristics, on page 30](#)
- [Feature History for Configuring Interface Characteristics, on page 30](#)

Information About Interface Characteristics

The following sections provide information about interface characteristics.

Interface Types

This section describes the different types of interfaces supported by the device. The rest of the chapter describes configuration procedures for physical interface characteristics.



Note The stack ports on the rear of the stacking-capable devices are not Ethernet ports and cannot be configured.

Port-Based VLANs

A VLAN is a switched network that is logically segmented by function, team, or application, without regard to the physical location of the users. Packets received on a port are forwarded only to ports that belong to the same VLAN as the receiving port. Network devices in different VLANs cannot communicate with one another without a Layer 3 device to route traffic between the VLANs.

VLAN partitions provide hard firewalls for traffic in the VLAN, and each VLAN has its own MAC address table. A VLAN comes into existence when a local port is configured to be associated with the VLAN, when the VLAN Trunking Protocol (VTP) learns of its existence from a neighbor on a trunk, or when a user creates a VLAN. VLANs can be formed with ports across the stack.

To configure VLANs, use the **vlan *vlan-id*** global configuration command to enter VLAN configuration mode. The VLAN configurations for normal-range VLANs (VLAN IDs 1 to 1005) are saved in the VLAN database. If VTP is version 1 or 2, to configure extended-range VLANs (VLAN IDs 1006 to 4094), you must first set VTP mode to transparent. Extended-range VLANs created in transparent mode are not added to the VLAN database but are saved in the running configuration. With VTP version 3, you can create extended-range

VLANs in client or server mode in addition to transparent mode. These VLANs are saved in the VLAN database.

In a switch stack, the VLAN database is downloaded to all switches in a stack, and all switches in the stack build the same VLAN database. The running configuration and the saved configuration are the same for all switches in a stack.

Add ports to a VLAN by using the **switchport** command in interface configuration mode.

- Identify the interface.
- For a trunk port, set trunk characteristics, and, if desired, define the VLANs to which it can belong.
- For an access port, set and define the VLAN to which it belongs.

Switch Ports

Switch ports are Layer 2-only interfaces associated with a physical port. Switch ports belong to one or more VLANs. A switch port can be an access port or a trunk port. You can configure a port as an access port or trunk port or let the Dynamic Trunking Protocol (DTP) operate on a per-port basis to set the switchport mode by negotiating with the port on the other end of the link. Switch ports are used for managing the physical interface and associated Layer 2 protocols and do not handle routing or bridging.

Configure switch ports by using the **switchport** interface configuration commands.

Access Ports

An access port belongs to and carries the traffic of only one VLAN (unless it is configured as a voice VLAN port). Traffic is received and sent in native formats with no VLAN tagging. Traffic arriving on an access port is assumed to belong to the VLAN assigned to the port. If an access port receives a tagged packet (Inter-Switch Link [ISL] or IEEE 802.1Q tagged), the packet is dropped, and the source address is not learned.

The types of access ports supported are:

- Static access ports are manually assigned to a VLAN (or through a RADIUS server for use with IEEE 802.1x).

You can also configure an access port with an attached Cisco IP Phone to use one VLAN for voice traffic and another VLAN for data traffic from a device attached to the phone.

Trunk Ports

A trunk port carries the traffic of multiple VLANs and by default is a member of all VLANs in the VLAN database. The IEEE 802.1Q trunk port type is supported. An IEEE 802.1Q trunk port supports simultaneous tagged and untagged traffic. An IEEE 802.1Q trunk port is assigned a default port VLAN ID (PVID), and all untagged traffic travels on the port default PVID. All untagged traffic and tagged traffic with a NULL VLAN ID are assumed to belong to the port default PVID. A packet with a VLAN ID equal to the outgoing port default PVID is sent untagged. All other traffic is sent with a VLAN tag.

Although by default, a trunk port is a member of every VLAN known to the VTP, you can limit VLAN membership by configuring an allowed list of VLANs for each trunk port. The list of allowed VLANs does not affect any other port but the associated trunk port. By default, all possible VLANs (VLAN ID 1 to 4094) are in the allowed list. A trunk port can become a member of a VLAN only if VTP knows of the VLAN and if the VLAN is in the enabled state. If VTP learns of a new, enabled VLAN and the VLAN is in the allowed list for a trunk port, the trunk port automatically becomes a member of that VLAN and traffic is forwarded to and from the trunk port for that VLAN. If VTP learns of a new, enabled VLAN that is not in the allowed

list for a trunk port, the port does not become a member of the VLAN, and no traffic for the VLAN is forwarded to or from the port.

Tunnel Ports

Tunnel ports are used in IEEE 802.1Q tunneling to segregate the traffic of customers in a service-provider network from other customers who are using the same VLAN number. You configure an asymmetric link from a tunnel port on a service-provider edge switch to an IEEE 802.1Q trunk port on the customer switch. Packets entering the tunnel port on the edge switch, already IEEE 802.1Q-tagged with the customer VLANs, are encapsulated with another layer of an IEEE 802.1Q tag (called the metro tag), containing a VLAN ID unique in the service-provider network, for each customer. The double-tagged packets go through the service-provider network keeping the original customer VLANs separate from those of other customers. At the outbound interface, also a tunnel port, the metro tag is removed, and the original VLAN numbers from the customer network are retrieved.

Tunnel ports cannot be trunk ports or access ports and must belong to a VLAN unique to each customer.

Routed Ports

A routed port is a physical port that acts like a port on a router; it does not have to be connected to a router. A routed port is not associated with a particular VLAN, as is an access port. A routed port behaves like a regular router interface, except that it does not support VLAN subinterfaces. Routed ports can be configured with a Layer 3 routing protocol. A routed port is a Layer 3 interface only and does not support Layer 2 protocols, such as DTP and STP.

Configure routed ports by putting the interface into Layer 3 mode with the **no switchport** interface configuration command. Then assign an IP address to the port, enable routing, and assign routing protocol characteristics by using the **ip routing** and **router protocol** global configuration commands.



Note Entering a **no switchport** interface configuration command shuts down the interface and then re-enables it, which might generate messages on the device to which the interface is connected. When you put an interface that is in Layer 2 mode into Layer 3 mode, the previous configuration information related to the affected interface might be lost.



Note A port configured as a switchport does not support MAC address configuration. It does not support the **mac-address x.x.x** command.

The number of routed ports that you can configure is not limited by software. However, the interrelationship between this number and the number of other features being configured might impact CPU performance because of hardware limitations.

Switch Virtual Interfaces

A switch virtual interface (SVI) represents a VLAN of switch ports as one interface to the routing function in the system. You can associate only one SVI with a VLAN. You configure an SVI for a VLAN only to route between VLANs or to provide IP host connectivity to the device. By default, an SVI is created for the default VLAN (VLAN 1) to permit remote device administration. Additional SVIs must be explicitly configured.



Note You cannot delete interface VLAN 1.

SVIs provide IP host connectivity only to the system. SVIs are created the first time that you enter the **vlan** interface configuration command for a VLAN interface. The VLAN corresponds to the VLAN tag associated with data frames on an ISL or IEEE 802.1Q encapsulated trunk or the VLAN ID configured for an access port. Configure a VLAN interface for each VLAN for which you want to route traffic, and assign it an IP address.

You can also use the interface range command to configure existing VLAN SVIs within the range. The commands entered under the interface range command are applied to all existing VLAN SVIs within the range. You can enter the command **interface range create vlan** *x - y* to create all VLANs in the specified range that do not already exist. When the VLAN interface is created, **interface range vlan** *id* can be used to configure the VLAN interface.

Although the device stack or standalone device supports a total of 1005 VLANs and SVIs, the interrelationship between the number of SVIs and routed ports and the number of other features being configured might impact CPU performance because of hardware limitations.

When you create an SVI, it does not become active until it is associated with a physical port.

EtherChannel Port Groups

EtherChannel port groups treat multiple switch ports as one switch port. These port groups act as a single logical port for high-bandwidth connections between devices or between devices and servers. An EtherChannel balances the traffic load across the links in the channel. If a link within the EtherChannel fails, traffic previously carried over the failed link changes to the remaining links. You can group multiple trunk ports into one logical trunk port, group multiple access ports into one logical access port, group multiple tunnel ports into one logical tunnel port, or group multiple routed ports into one logical routed port. Most protocols operate over either single ports or aggregated switch ports and do not recognize the physical ports within the port group. Exceptions are the DTP, the Cisco Discovery Protocol (CDP), and the Port Aggregation Protocol (PAgP), which operate only on physical ports.

When you configure an EtherChannel, you create a port-channel logical interface and assign an interface to the EtherChannel. For Layer 3 interfaces, you manually create the logical interface by using the **interface port-channel** global configuration command. Then you manually assign an interface to the EtherChannel by using the **channel-group** interface configuration command. For Layer 2 interfaces, use the **channel-group** interface configuration command to dynamically create the port-channel logical interface. This command binds the physical and logical ports together.

Network Modules

The following table shows the list of supported uplink ports:

Speed	C9200	C9200L
One-Gigabit Ethernet	—	Fixed uplink ports
10-Gigabit Ethernet	Modular uplink ports	Fixed uplink ports
25-Gigabit Ethernet	Modular uplink ports	Fixed uplink ports

If you need an ethernet connection, use GLC-TE copper SFP for one Gigabit Ethernet on all modules.

The following SFP, SFP+, SFP28, QSFP ports are supported:

- 4x1G (only C9200L)
- 4x10G (C9200 and C9200L)
- 2x25G (C9200 and C9200L)

Power over Ethernet

The Power over Ethernet (PoE) technology allows PoE (802.3af standard), PoE+ (802.3at), and PoE++ (802.3bt) ports to supply power for the operation of a device.

For more information, see the *Configuring PoE* section of this guide.

Using the Switch USB Ports

The device has two USB Type A ports on the front panel.

USB Mini-Type B Console Port

The device has the following console ports:

- USB mini-Type B console connection
- RJ-45 console port

Console output appears on devices connected to both ports, but console input is active on only one port at a time. By default, the USB connector takes precedence over the RJ-45 connector.



Note Windows PCs require a driver for the USB port. See the hardware installation guide for driver installation instructions.

Use the supplied USB Type A-to-USB mini-Type B cable to connect a PC or other device to the device. The connected device must include a terminal emulation application. When the device detects a valid USB connection to a powered-on device that supports host functionality (such as a PC), input from the RJ-45 console is immediately disabled, and input from the USB console is enabled. Removing the USB connection immediately reenables input from the RJ-45 console connection. An LED on the device shows which console connection is in use.

Console Port Change Logs

At software startup, a log shows whether the USB or the RJ-45 console is active. Every device always first displays the RJ-45 media type.

In the sample output, device 1 has a connected USB console cable. Because the bootloader did not change to the USB console, the first log from the device shows the RJ-45 console. A short time later, the console changes and the USB console log appears. device 2 and device 3 have connected RJ-45 console cables.

```
switch-stack-1
*Mar  1 00:01:00.171: %USB_CONSOLE-6-MEDIA_RJ45: Console media-type is RJ45.
*Mar  1 00:01:00.431: %USB_CONSOLE-6-MEDIA_USB: Console media-type is USB.
```

When the USB cable is removed or the PC de-activates the USB connection, the hardware automatically changes to the RJ-45 console interface:

You can configure the console type to always be RJ-45, and you can configure an inactivity timeout for the USB connector.

USB Type A Port

The USB Type A port provides access to external USB flash devices, also known as thumb drives or USB keys. The port supports Cisco USB flash drives with capacities from 128 MB to 16 GB (USB devices with port densities of 128 MB, 256 MB, 1 GB, 4 GB, 8 GB, and 16 GB are supported). You can use standard Cisco IOS command-line interface (CLI) commands to read, write, erase, and copy to or from the flash device. You can also configure the devices to boot from the USB flash drive.

Disabling USB Ports

From Cisco IOS XE Bengaluru 17.5.x, all the USB ports in a standalone or stacked device can be disabled using the **platform usb disable** command. Configuring this command disables all external media, including both USB flash and SD flash devices. To reenables the USB ports, use the **no platform usb disable** command.

When a USB port is disabled, no system messages are generated if a USB is inserted.

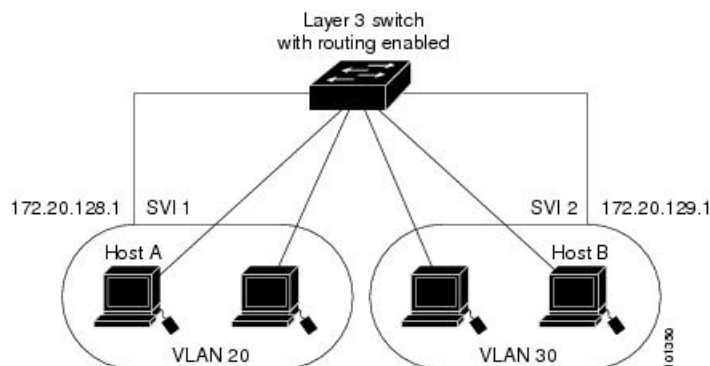


Note The **platform usb disable** command does not disable Bluetooth dongles connected to USB ports.

Interface Connections

Devices within a single VLAN can communicate directly through any switch. Ports in different VLANs cannot exchange data without going through a routing device. With a standard Layer 2 device, ports in different VLANs have to exchange information through a router. By using the device with routing enabled, when you configure both VLAN 20 and VLAN 30 with an SVI to which an IP address is assigned, packets can be sent from Host A to Host B directly through the device with no need for an external router.

Figure 1: Connecting VLANs with a Switch



When the Network Advantage license is used on the device or the active device, the device uses the routing method to forward traffic between interfaces. If the Network Essentials license is used on the device or the active device, only basic routing (static routing and RIP) is supported. Whenever possible, to maintain high

performance, forwarding is done by the device hardware. However, only IPv4 packets with Ethernet II encapsulation are routed in hardware.

The routing function can be enabled on all SVIs and routed ports. The device routes only IP traffic. When IP routing protocol parameters and address configuration are added to an SVI or routed port, any IP traffic received from these ports is routed.

Interface Configuration Mode

The device supports these interface types:

- Physical ports: Device ports and routed ports
- VLANs: Switch virtual interfaces
- Port channels: EtherChannel interfaces

You can also configure a range of interfaces.

To configure a physical interface (port), specify the interface type, stack member number (only stacking-capable switches), module number, and device port number, and enter interface configuration mode.

- Type: Gigabit Ethernet (GigabitEthernet or gi) for 10/100/1000 Mbps Ethernet ports, 10-Gigabit Ethernet (TenGigabitEthernet or te) for 10 Gbps, 25-Gigabit Ethernet (TwentyFiveGigE or twe) for 25 Gbps, and small form-factor pluggable (SFP) module Gigabit Ethernet interfaces.
- You can use the switch port LEDs in Stack mode to identify the stack member number of a device.
- Module number: The module or slot number on the device: switch (downlink) ports are 0, and uplink ports are 1.
- On a device with SFP uplink ports, the module number is 1 and the port numbers restart. For example, if the device has 24 10/100/1000 ports, the SFP module ports are GigabitEthernet1/1/1 through GigabitEthernet1/1/4 or TenGigabitEthernet1/1/1 through TenGigabitEthernet1/1/4.

You can identify physical interfaces by physically checking the interface location on the device. You can also use the **show** privileged EXEC commands to display information about a specific interface or all the interfaces on the switch. The remainder of this chapter primarily provides physical interface configuration procedures.

These are examples of how to configure interfaces on stacking-capable and standalone device:

- To configure 10/100/1000 port 4 on a standalone device, enter this command:

```
Device# configure terminal  
Device(config)# interface GigabitEthernet1/0/4
```

- To configure 10-Gigabit Ethernet port 1 on a standalone device, enter this command:

```
Device# configure terminal  
Device(config)# interface TenGigabitEthernet 1/1/1
```

- To configure 10-Gigabit Ethernet port on stack member 3, enter this command:

```
Device# configure terminal  
Device(config)# interface TenGigabitEthernet 3/1/1
```

- To configure the first SFP module (uplink) port on a standalone device, enter this command:

```
Device# configure terminal
Device(config)# interface GigabitEthernet 1/1/1
```

Default Ethernet Interface Configuration

To configure Layer 2 parameters, if the interface is in Layer 3 mode, you must enter the **switchport** interface configuration command without any parameters to put the interface into Layer 2 mode. This shuts down the interface and then re-enables it, which might generate messages on the device to which the interface is connected. When you put an interface that is in Layer 3 mode into Layer 2 mode, the previous configuration information related to the affected interface might be lost, and the interface is returned to its default configuration.

This table shows the Ethernet interface default configuration, including some features that apply only to Layer 2 interfaces.

Table 1: Default Layer 2 Ethernet Interface Configuration

Feature	Default Setting
Operating mode	Layer 2 or switching mode (switchport command).
Allowed VLAN range	VLANs 1 to 4094.
Default VLAN (for access ports)	VLAN 1 (Layer 2 interfaces only).
Native VLAN (for IEEE 802.1Q trunks)	VLAN 1 (Layer 2 interfaces only).
VLAN trunking	Switchport mode dynamic auto (supports DTP) (Layer 2 interfaces only).
Port enable state	All ports are enabled.
Port description	None defined.
Speed	Autonegotiate. Autonegotiation is supported on 25G, 40G, and 100G CU cables on the Cisco Catalyst 9500 Series Switches - High Performance. On Cisco Catalyst 9500X series switches, autonegotiation is enabled on 25G, 50G, 40G, 100G and 400G copper links, by default. Their speed is determined by the type of transceiver module plugged in.
Duplex mode	Autonegotiate.
Flow control	Flow control is set to receive: on . It is always off for sent packets.
EtherChannel (PAgP)	Disabled on all Ethernet ports.
Port blocking (unknown multicast and unknown unicast traffic)	Disabled (not blocked) (Layer 2 interfaces only).
Broadcast, multicast, and unicast storm control	Disabled.

Feature	Default Setting
Protected port	Disabled (Layer 2 interfaces only).
Port security	Disabled (Layer 2 interfaces only).
Port Fast	Disabled.
Auto-MDIX	Enabled. Note The switch might not support a pre-standard powered device, such as Cisco IP phones and access points that do not fully support IEEE 802.3af, if that powered device is connected to the switch through a crossover cable. This is regardless of whether auto-MIDX is enabled on the switch port.
Power over Ethernet (PoE)	Enabled (auto).

Interface Speed and Duplex Mode

Gigabit Ethernet interfaces on the switch operate at 10, 100, 1000 Mbps speed and in either full-duplex or half-duplex mode. In full-duplex mode, two stations can send and receive traffic at the same time. Normally, 10-Mbps ports operate in half-duplex mode, which means that stations can either receive or send traffic. The switch also includes multi-Gigabit Ethernet ports, which support speeds at 100 Mb, 1 Gb, 2.5 Gb, 5 Gb, and 10 Gb and operate at full-duplex mode, SFP modules that support speeds up to 1 Gbps, SFP+ modules that support speeds up to 10 Gbps, and QSFP modules that support speeds up to 40 Gb/s. For the list of supported switch models, refer *Cisco Catalyst 9200 Series Switches Hardware Installation Guide*.

Speed and Duplex Configuration Guidelines

When configuring an interface speed and duplex mode, note these guidelines:

- Gigabit Ethernet (10/100/1000 Mbps) ports support all speed options and all duplex options (auto, half, and full). However, Gigabit Ethernet ports operating at 1000 Mbps and above do not support half-duplex mode.

Multi-Gigabit Ethernet ports (100 Mbps, 1 Gbps, 2.5 Gbps, 5Gbps, 10 Gbps, 100 Gbps) support all speed options, but only support auto and full duplex mode. These ports do not support half-duplex mode at any speed.

SFP ports operating at 1 Gbps, SFP+ ports operating at 10 Gbps, and SFP28 ports operating at 25 Gbps and QSFP ports operating at 40 Gbps support only the **no speed non negotiate** or **speed non negotiate** commands. Duplex options are not supported.



Note SFP, SFP+, and SFP28 ports support speed (auto, 10, 100, 1000) and duplex (auto/full/half) options only if the 1000Base-T SFP is used. SFP, SFP+, and SFP28 ports support speed (auto/100) and duplex (auto/full/half) options only if the GLC-GE-100FX modules are used.

- If both ends of the line support autonegotiation, we highly recommend the default setting of **auto** negotiation.
- If one interface supports autonegotiation and the other end does not, configure duplex and speed on both interfaces; do not use the **auto** setting on the supported side.
- When STP is enabled and a port is reconfigured, the device can take up to 30 seconds to check for loops. The port LED is amber while STP reconfigures. As best practice, we suggest configuring the speed and duplex options on a link to **auto** or to **fixed** on both the ends. If one side of the link is configured to **auto** and the other side is configured to **fixed**, the link will not be up; this is expected behavior.
- As best practice, we recommend that you configure the speed and duplex options on a link to **auto** or to **fixed** on both the ends. If one side of the link is configured to **auto** and the other side is configured to **fixed**, the link will not be up; this is expected behavior.

**Caution**

Changing the interface speed and duplex mode configuration might shut down and re-enable the interface during the reconfiguration.

IEEE 802.3x Flow Control

Flow control enables connected Ethernet ports to control traffic rates during congestion by allowing congested nodes to pause link operation at the other end. If one port experiences congestion and cannot receive any more traffic, it notifies the other port by sending a pause frame to stop sending until the condition clears. Upon receipt of a pause frame, the sending device stops sending any data packets, which prevents any loss of data packets during the congestion period.

**Note**

The switch ports can receive, but not send, pause frames.

You use the **flowcontrol** interface configuration command to set the interface's ability to **receive** pause frames to **on**, **off**, or **desired**. The default state is **on**.

When set to **desired**, an interface can operate with an attached device that is required to send flow-control packets or with an attached device that is not required to but can send flow-control packets.

These rules apply to flow control settings on the device:

- **receive on** (or **desired**): The port cannot send pause frames but can operate with an attached device that is required to or can send pause frames; the port can receive pause frames.
- **receive off**: Flow control does not operate in either direction. In case of congestion, no indication is given to the link partner, and no pause frames are sent or received by either device.

**Note**

For details on the command settings and the resulting flow control resolution on local and remote ports, see the **flowcontrol** interface configuration command in the command reference for this release.

Layer 3 Interfaces

The device supports these types of Layer 3 interfaces:

- SVIs: You should configure SVIs for any VLANs for which you want to route traffic. SVIs are created when you enter a VLAN ID following the **interface vlan** global configuration command. To delete an SVI, use the **no interface vlan** global configuration command. You cannot delete interface VLAN 1.



Note When you create an SVI, it does not become active until it is associated with a physical port.

When configuring SVIs, you can use the **switchport autostate exclude** command on a port to exclude that port from being included in determining SVI line-state. To disable autostate on the SVI, use the **no autostate** command on the SVI.

- Routed ports: Routed ports are physical ports configured to be in Layer 3 mode by using the **no switchport** interface configuration command.
- Layer 3 EtherChannel ports: EtherChannel interfaces made up of routed ports.

A Layer 3 device can have an IP address assigned to each routed port and SVI.

There is no defined limit to the number of SVIs and routed ports that can be configured in a device or in a device stack. However, the interrelationship between the number of SVIs and routed ports and the number of other features being configured might have an impact on CPU usage because of hardware limitations. If the device is using its maximum hardware resources, attempts to create a routed port or SVI have these results:

- If you try to create a new routed port, the device generates a message that there are not enough resources to convert the interface to a routed port, and the interface remains as a switchport.
- If you try to create an extended-range VLAN, an error message is generated, and the extended-range VLAN is rejected.
- If the device is notified by VLAN Trunking Protocol (VTP) of a new VLAN, it sends a message that there are not enough hardware resources available and shuts down the VLAN. The output of the **show vlan** user EXEC command shows the VLAN in a suspended state.
- If the device attempts to boot up with a configuration that has more VLANs and routed ports than hardware can support, the VLANs are created, but the routed ports are shut down, and the device sends a message that this was due to insufficient hardware resources.



Note All Layer 3 interfaces require an IP address to route traffic. This procedure shows how to configure an interface as a Layer 3 interface and how to assign an IP address to an interface:

If the physical port is in Layer 2 mode (the default), you must enter the **no switchport** interface configuration command to put the interface into Layer 3 mode. Entering a **no switchport** command disables and then re-enables the interface, which might generate messages on the device to which the interface is connected. Furthermore, when you put an interface that is in Layer 2 mode into Layer 3 mode, the previous configuration information related to the affected interface might be lost, and the interface is returned to its default configuration.

How to Configure Interface Characteristics

The following sections provide information about the various tasks that comprise the procedure to configure interface characteristics.

Configuring an Interface

These general instructions apply to all interface configuration processes.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface Example: Device(config)# interface gigabitethernet1/0/1 Device(config-if)#	Identifies the interface type, and the number of the connector. Note You do not need to add a space between the interface type and the interface number. For example, in the preceding line, you can specify either gigabitethernet 1/0/1 , gigabitethernet1/0/1 , gi 1/0/1 , or gi1/0/1 .
Step 4	Follow each interface command with the interface configuration commands that the interface requires.	Defines the protocols and applications that will run on the interface. The commands are collected and applied to the interface when you enter another interface command or enter end to return to privileged EXEC mode.
Step 5	interface range or interface range macro	(Optional) Configures a range of interfaces. Note Interfaces configured in a range must be the same type and must be configured with the same feature options.
Step 6	show interfaces	Displays a list of all interfaces on or configured for the switch. A report is provided for each interface that the device supports or for the specified interface.

Adding a Description for an Interface

Follow these steps to add a description for an interface.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet1/0/2	Specifies the interface for which you are adding a description, and enter interface configuration mode.
Step 4	description <i>string</i> Example: Device(config-if)# description Connects to Marketing	Adds a description for an interface.
Step 5	end Example: Device(config-if)# end	Returns to privileged EXEC mode.
Step 6	show interfaces <i>interface-id</i> description	Verifies your entry.
Step 7	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring a Range of Interfaces

To configure multiple interfaces with the same configuration parameters, use the **interface range** global configuration command. When you enter the interface-range configuration mode, all command parameters that you enter are attributed to all interfaces within that range until you exit this mode.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface range { <i>port-range</i> macro <i>macro_name</i> } Example: Device (config)# interface range macro	Specifies the range of interfaces (VLANs or physical ports) to be configured, and enter interface-range configuration mode. <ul style="list-style-type: none"> You can use the interface range command to configure up to five port ranges or a previously defined macro. The macro variable is explained in Configuring and Using Interface Range Macros. In a comma-separated <i>port-range</i>, you must enter the interface type for each entry and enter spaces before and after the comma. In a hyphen-separated <i>port-range</i>, you do not need to re-enter the interface type, but you must enter a space before the hyphen. <p>Note Use the normal configuration commands to apply the configuration parameters to all interfaces in the range. Each command is executed as it is entered.</p>
Step 4	end Example: Device (config)# end	Returns to privileged EXEC mode.
Step 5	show interfaces [<i>interface-id</i>] Example: Device# show interfaces	Verifies the configuration of the interfaces in the range.

	Command or Action	Purpose
Step 6	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring and Using Interface Range Macros

You can create an interface range macro to automatically select a range of interfaces for configuration. Before you can use the **macro** keyword in the **interface range macro** global configuration command string, you must use the **define interface-range** global configuration command to define the macro.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <pre>Device> enable</pre>	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	define interface-range <i>macro_name</i> <i>interface-range</i> Example:	Defines the interface-range macro, and saves it in NVRAM. <ul style="list-style-type: none"> • The <i>macro_name</i> is a 32-character maximum character string. • A macro can contain up to five comma-separated interface ranges. • Each <i>interface-range</i> must consist of the same port type. <p>Note Before you can use the macro keyword in the interface range macro global configuration command string, you must use the define interface-range global configuration command to define the macro.</p>

	Command or Action	Purpose
Step 4	interface range macro <i>macro_name</i> Example: Device(config)# interface range macro enet_list	Selects the interface range to be configured using the values saved in the interface-range macro called <i>macro_name</i> . You can now use the normal configuration commands to apply the configuration to all interfaces in the defined macro.
Step 5	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 6	show running-config include define Example: Device# show running-config include define	Shows the defined interface range macro configuration.
Step 7	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Setting the Interface Speed and Duplex Parameters

Follow these steps to configure the interface speed and duplex parameters.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet1/0/3	Specifies the physical interface to be configured, and enters interface configuration mode.

	Command or Action	Purpose
Step 4	duplex {auto full half} Example: Device(config-if)# duplex half	Enters the duplex parameter for the interface. Enables half-duplex mode (for interfaces operating only at 10 or 100 Mb/s). Half duplex is not supported on multi-Gigabit Ethernet ports configured for speed of 1000 Mb/s. You can configure the duplex setting when the speed is set to auto .
Step 5	end Example: Device(config-if)# end	Returns to privileged EXEC mode.
Step 6	show interfaces interface-id Example: Device# show interfaces gigabitethernet1/0/3	Displays the interface speed and duplex mode configuration.
Step 7	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring the IEEE 802.3x Flow Control

Follow these steps to configure the IEEE 802.3x flow control.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode
Step 3	interface interface-id Example:	Specifies the physical interface to be configured, and enters interface configuration mode.

	Command or Action	Purpose
	Device(config)# interface gigabitethernet1/0/1	
Step 4	flowcontrol {receive} {on off desired} Example: Device(config-if)# flowcontrol receive on	Configures the flow control mode for the port.
Step 5	end Example: Device(config-if)# end	Returns to privileged EXEC mode.
Step 6	show flowcontrol interface <i>interface-id</i> Example: Device# show flowcontrol interface GigabitEthernet1/0/1	Verifies the specified interface flow control settings.
Step 7	show flowcontrol module <i>slot</i> Example: Device# show flowcontrol module 1	Verifies the interface flow control settings on the module.
Step 8	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring a Layer 3 Interface

Follow these steps to configure a layer 3 interface.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example:	Enters global configuration mode.

	Command or Action	Purpose
	Device# <code>configure terminal</code>	
Step 3	interface { gigabitethernet <i>interface-id</i> } { vlan <i>vlan-id</i> } { port-channel <i>port-channel-number</i> } Example: Device(config)# interface gigabitethernet1/0/2	Specifies the interface to be configured as a Layer 3 interface, and enters interface configuration mode.
Step 4	no switchport Example: Device(config-if)# no switchport	(For physical ports only) Enters Layer 3 mode.
Step 5	ip address <i>ip_address subnet_mask</i> Example: Device(config-if)# ip address 192.20.135.21 255.255.255.0	Configures the IP address and IP subnet.
Step 6	no shutdown Example: Device(config-if)# no shutdown	Enables the interface.
Step 7	end Example: Device(config-if)# end	Returns to privileged EXEC mode.
Step 8	show interfaces [<i>interface-id</i>]	Verifies the configuration.
Step 9	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring a Logical Layer 3 GRE Tunnel Interface

Before you begin

Generic Routing Encapsulation (GRE) is a tunneling protocol used to encapsulate network layer protocols inside virtual point-to-point links. A GRE tunnel only provides encapsulation and not encryption.

**Note**

- GRE tunnels are supported on the hardware on Cisco Catalyst 9000 switches. When GRE is configured without tunnel options, packets are hardware-switched. When GRE is configured with tunnel options (such as key, checksum, and so on), packets are switched in the software. A maximum of 10 GRE tunnels are supported.
- Other features such as Access Control Lists (ACL) and Quality of Service (QoS) are not supported for the GRE tunnels.
- The **tunnel path-mtu-discovery** command is not supported for GRE tunnels. To avoid fragmentation, you can set the maximum transmission unit (MTU) of both ends of the GRE tunnel to the lowest value by using the **ip mtu 256** command.

To configure a GRE tunnel, perform this task:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface tunnel number Example: Device(config)# interface tunnel 2	Enables tunneling on the interface.
Step 4	ip address ip_address subnet_mask Example: Device(config)# ip address 100.1.1.1 255.255.255.0	Configures the IP address and IP subnet.
Step 5	tunnel source {ip_address type_number} Example: Device(config)# tunnel source 10.10.10.1	Configures the tunnel source.
Step 6	tunnel destination {host_name ip_address} Example: Device(config)# tunnel destination 10.10.10.2	Configures the tunnel destination.
Step 7	tunnel mode gre ip Example:	Configures the tunnel mode.

	Command or Action	Purpose
	Device(config)# tunnel mode gre ip	
Step 8	end Example: Device(config)# end	Exits configuration mode.

Configuring SVI Autostate Exclude

Follow these steps to exclude SVI autostate.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet1/0/2	Specifies a Layer 2 interface (physical port or port channel), and enters interface configuration mode.
Step 4	switchport autostate exclude Example: Device(config-if)# switchport autostate exclude	Excludes the access or trunk port when defining the status of an SVI line state (up or down)
Step 5	end Example: Device(config-if)# end	Returns to privileged EXEC mode.
Step 6	show running config interface <i>interface-id</i> Example: Device# show running config interface gigabitethernet1/0/2	(Optional) Shows the running configuration. Verifies the configuration.

	Command or Action	Purpose
Step 7	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Shutting Down and Restarting an Interface

Shutting down an interface disables all functions on the specified interface and marks the interface as unavailable on all monitoring command displays. This information is communicated to other network servers through all dynamic routing protocols. The interface is not mentioned in any routing updates.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface {vlan vlan-id} { gigabitethernet interface-id} {port-channel port-channel-number} Example: Device(config)# interface gigabitethernet1/0/2	Selects the interface to be configured.
Step 4	shutdown Example: Device(config-if)# shutdown	Shuts down an interface.
Step 5	no shutdown Example: Device(config-if)# no shutdown	Restarts an interface.
Step 6	end Example:	Returns to privileged EXEC mode.

	Command or Action	Purpose
	<code>Device(config-if)# end</code>	
Step 7	show running-config Example: <code>Device# show running-config</code>	Verifies your entries.

Configuring the Console Media Type

Follow these steps to set the console media type to RJ-45. If you configure the console as RJ-45, USB console operation is disabled, and input comes only through the RJ-45 connector.

Procedure

	Command or Action	Purpose
Step 1	enable Example: <code>Device> enable</code>	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: <code>Device# configure terminal</code>	Enters global configuration mode.
Step 3	line console 0 Example: <code>Device(config)# line console 0</code>	Configures the console and enters line configuration mode.
Step 4	media-type rj45 switch <i>switch_number</i> Example: <code>Device(config-line)# media-type rj45 switch 1</code>	Configures the console media type to be only RJ-45 port. If you do not enter this command and both types are connected, the USB port is used by default.
Step 5	end Example: <code>Device(config)# end</code>	Returns to privileged EXEC mode.
Step 6	copy running-config startup-config Example: <code>Device# copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Configuring USB Inactivity Timeout

When the USB console port is deactivated due to a timeout, you can restore its operation by disconnecting and reconnecting the USB cable.



Note The configured inactivity timeout applies to all device in a stack. However, a timeout on one device does not cause a timeout on other device in the stack.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	line console 0 Example: Device(config)# line console 0	Configures the console and enters line configuration mode.
Step 4	usb-inactivity-timeout switch <i>switch_number</i> <i>timeout-minutes</i> Example: Device(config-line)# usb-inactivity-timeout switch 1 30	Specifies an inactivity timeout for the console port. The range is 1 to 240 minutes. The default is to have no timeout configured.
Step 5	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Disabling USB Ports

To disable all USB ports, perform this procedure.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	[no] platform usb disable Example: Device(config)# platform usb disable	Disables all the USB ports on the device. Use the no platform usb disable command to reenabling the USB ports.
Step 4	exit Example: Device(config)# exit	Exits to privileged EXEC mode.
Step 5	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Monitoring Interface Characteristics

The following sections provide information about monitoring interface characteristics.

Monitoring Interface Status

Commands entered at the privileged EXEC prompt display information about the interface, including the versions of the software and the hardware, the configuration, and statistics about the interfaces.

Table 2: show Commands for Interfaces

Command	Purpose
show interfaces <i>interface-id</i> status [err-disabled]	Displays interface status or a list of interfaces in the error-disabled state.
show interfaces [<i>interface-id</i>] switchport	Displays administrative and operational status of switching (nonrouting) ports. You can use this command to find out if a port is in routing or in switching mode.

Command	Purpose
show interfaces [<i>interface-id</i>] description	Displays the description configured on an interface or all interfaces and the interface status.
show ip interface [<i>interface-id</i>]	Displays the usability status of all interfaces configured for IP routing or the specified interface.
show interface [<i>interface-id</i>] stats	Displays the input and output packets by the switching path for the interface.
show interface [<i>interface-id</i>] link [module number]	Displays the up time and down time of an interface or all interfaces.
show interfaces <i>interface-id</i>	(Optional) Displays speed and duplex on the interface.
show interfaces transceiver dom-supported-list	(Optional) Displays Digital Optical Monitoring (DOM) status on the connect SFP modules.
show interfaces transceiver properties	(Optional) Displays temperature, voltage, or amount of current on the interface.
show interfaces [<i>interface-id</i>] [{ transceiver properties detail }] <i>module number</i>	Displays physical and operational status about an SFP module.
show running-config interface [<i>interface-id</i>]	Displays the running configuration in RAM for the interface.
show version	Displays the hardware configuration, software version, the names and sources of configuration files, and the boot images.
show controllers ethernet-controller <i>interface-id</i> phy	Displays the operational state of the auto-MDIX feature on the interface.

Clearing and Resetting Interfaces and Counters

Table 3: clear Commands for Interfaces

Command	Purpose
clear counters [<i>interface-id</i>]	Clears interface counters.
clear interface <i>interface-id</i>	Resets the hardware logic on an interface.
clear line [<i>number</i> console 0 vty number]	Resets the hardware logic on an asynchronous serial line.



Note The **clear counters** privileged EXEC command does not clear counters retrieved by using Simple Network Management Protocol (SNMP), but only those seen with the **show interface** privileged EXEC command.

Configuration Examples for Interface Characteristics

The following sections provide examples of interface characteristics configurations.

Example: Adding a Description to an Interface

The following example shows how to add a description to an interface:

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTRL/Z.
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# description Connects to Marketing
Device(config-if)# end
Device# show interfaces gigabitethernet1/0/2 description

Interface Status          Protocol Description
Gi1/0/2   admin down        down    Connects to Marketing
```

Example: Configuring Interfaces on a Stack-Capable Switch

The following example shows how to configure 10/100/1000 port 4 on a standalone switch:

```
Device(config)# interface gigabitethernet1/1/4
```

The following example shows how to configure the first SFP module uplink port on stack member 1:

```
Device(config)# interface gigabitethernet1/1/1
```

The following example shows how to configure 10-Gigabit Ethernet port on stack member 3:

```
Device(config)# interface tengigabitethernet3/0/1
```

Example: Configuring a Range of Interfaces

The following example shows how to use the **interface range** global configuration command to set the speed to 100 Mb/s on ports 1 to 4 on switch 1:

```
Device# configure terminal
Device(config)# interface range gigabitethernet1/0/1 - 4
Device(config-if-range)# speed 100
```

The following example shows how to use a comma to add different interface type strings to the range to enable Gigabit Ethernet ports 1 to 3 and 10-Gigabit Ethernet ports 1 and 2 to receive flow-control pause frames:

```
Device# configure terminal
Device(config)# interface range gigabitethernet1/1/1 - 3 , tengigabitethernet1/1/1 - 2
Device(config-if-range)# flowcontrol receive on
```



Note If you enter multiple configuration commands while you are in interface-range mode, each command is executed as it is entered. The commands are not batched and executed after you exit interface-range mode. If you exit interface-range configuration mode while the commands are being executed, some commands might not be executed on all interfaces in the range. Wait until the command prompt reappears before exiting interface-range configuration mode.

Example: Configuring and Using Interface Range Macros

The following example shows how to enter interface-range configuration mode for the interface-range macro *enet_list*:

```
Device# configure terminal
Device(config)# interface range macro enet_list
Device(config-if-range)#
```

The following example shows how to delete the interface-range macro *enet_list* and to verify that it was deleted:

```
Device# configure terminal
Device(config)# no define interface-range enet_list
Device(config)# end
Device# show run | include define
Device#
```

Example: Setting Interface Speed and Duplex Mode

The following example shows how to set the interface speed to 10 Mbps and the duplex mode to full, on a 10/100/1000 Mbps port:

```
Device# configure terminal
Device(config)# interface gigabitethernet1/0/3
Device(config-if)# speed 10
Device(config-if)# duplex full
```

The following example shows how to set the interface speed to 100 Mbps on a 10/100/1000 Mbps port:

```
Device# configure terminal
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# speed 100
```

Example: Configuring a Layer 3 Interface

The following example shows how to configure a Layer 3 interface:

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# no switchport
Device(config-if)# ip address 192.20.135.21 255.255.255.0
```

```
Device(config-if)# no shutdown
```

Example: Configuring the Console Media Type

The following example shows how to disable the USB console media type and enable the RJ-45 console media type:

```
Device# configure terminal
Device(config)# line console 0
Device(config-line)# media-type rj45 switch 1
```

This configuration terminates any active USB console media type in the stack. A log shows that this termination has occurred. This example shows that the console on switch 1 reverted to RJ-45.

```
*Mar 1 00:25:36.860: %USB_CONSOLE-6-CONFIG_DISABLE: Console media-type USB disabled by
system configuration, media-type reverted to RJ45.
```

At this point no switches in the stack allow a USB console to have input. A log entry shows when a console cable is attached. If a USB console cable is connected to switch 2, it is prevented from providing input.

```
*Mar 1 00:34:27.498: %USB_CONSOLE-6-CONFIG_DISALLOW: Console media-type USB is disallowed
by system configuration, media-type remains RJ45. (switch-stk-2)
```

The following example shows how to reverse the previous configuration and immediately activate any USB console that is connected:

```
Device# configure terminal
Device(config)# line console 0
Device(config-line)# no media-type rj45 switch 1
```

Example: Configuring USB Inactivity Timeout

The following example shows how to configure the inactivity timeout to 30 minutes:

```
Device# configure terminal
Device(config)# line console 0
Device(config-line)# usb-inactivity-timeout switch 1 30
```

The following example shows how to disable the configuration:

```
Device# configure terminal
Device(config)# line console 0
Device(config-line)# no usb-inactivity-timeout switch 1
```

If there is no (input) activity on a USB console port for the configured number of minutes, the inactivity timeout setting applies to the RJ-45 port, and a log shows this occurrence:

```
*Mar 1 00:47:25.625: %USB_CONSOLE-6-INACTIVITY_DISABLE: Console media-type USB disabled
due to inactivity, media-type reverted to RJ45.
```

At this point, the only way to reactivate the USB console port is to disconnect and reconnect the cable.

When the USB cable on a switch is disconnected and reconnected, a log, which is similar to this, appears:

```
*Mar 1 00:48:28.640: %USB_CONSOLE-6-MEDIA_USB: Console media-type is USB.
```

Additional References for Configuring Interface Characteristics

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the "Interface and Hardware Commands" section in the <i>Command Reference (Catalyst 9200 Series Switches)</i> .

Feature History for Configuring Interface Characteristics

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Fuji 16.9.2	Interface Characteristics	Interface Characteristics includes interface types, connections, configuration modes, speed, and other aspects of configuring a physical interface on a device.
Cisco IOS XE Gibraltar 16.11.1	Multi-Gigabit Ethernet Interfaces	Support for Multi-Gigabit Ethernet ports operating at 100Mb/s, 1Gb/s, 2.5 Gb/s, 5Gb/s, and 10 Gb/s was introduced on all the models of the series
Cisco IOS XE Bengaluru 17.5.1	Disabling USB interfaces	Support to disable all USB ports on a standalone or stacked device was introduced.
Cisco IOS XE Cupertino 17.9.1	Interface Characteristics Disabling USB interfaces Multi-Gigabit Ethernet Interfaces	This feature was implemented on C9200CX-12P-2X2G, C9200CX-8P-2X2G, and C9200CX-12T-2X2G models of the Cisco Catalyst 9200CX Series Switches, which were introduced in this release.

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to [Cisco Feature Navigator](#).



CHAPTER 2

Configuring Auto-MDIX

- [Prerequisites for Auto-MDIX, on page 31](#)
- [Restrictions for Auto-MDIX, on page 31](#)
- [Information About Configuring Auto-MDIX, on page 31](#)
- [How to Configure Auto-MDIX, on page 32](#)
- [Example for Configuring Auto-MDIX, on page 33](#)
- [Auto-MDIX and Operational State, on page 33](#)
- [Additional References for Auto-MDIX, on page 34](#)
- [Feature History for Auto-MDIX, on page 34](#)

Prerequisites for Auto-MDIX

To configure Layer 2 parameters, if the interface is in Layer 3 mode, you must enter the **switchport** interface configuration command without any parameters to put the interface into Layer 2 mode. This shuts down the interface and then re-enables it, which might generate messages on the device to which the interface is connected. When you put an interface that is in Layer 3 mode into Layer 2 mode, the previous configuration information related to the affected interface might be lost, and the interface is returned to its default configuration.

Automatic medium-dependent interface crossover (auto-MDIX) is enabled by default.

Restrictions for Auto-MDIX

- The device might not support a pre-standard powered device—such as Cisco IP phones and access points that do not fully support IEEE 802.3af—if that powered device is connected to the device through a crossover cable. This is regardless of whether auto-MIDX is enabled on the switch port.
- After each device reload, interfaces configured with the **no mdix auto** command will be in down state. To enable the interface, each time after a reload, you have to remove the SFP and reinsert the SFP.

Information About Configuring Auto-MDIX

The following sections provide information about Auto-MDIX.

Auto-MDIX on an Interface

When automatic medium-dependent interface crossover (auto-MDIX) is enabled on an interface, the interface automatically detects the required cable connection type (straight through or crossover) and configures the connection appropriately. When connecting devices without the auto-MDIX feature, you must use straight-through cables to connect to devices such as servers, workstations, or routers and crossover cables to connect to other devices or repeaters. With auto-MDIX enabled, you can use either type of cable to connect to other devices, and the interface automatically corrects for any incorrect cabling. For more information about cabling requirements, see the hardware installation guide.



Note Auto-MDIX is enabled by default.

This table shows the link states that result from auto-MDIX settings and correct and incorrect cabling.

Table 4: Link Conditions and Auto-MDIX Settings

Local Side Auto-MDIX	Remote Side Auto-MDIX	With Correct Cabling	With Incorrect Cabling
On	On	Link up	Link up
On	Off	Link up	Link up
Off	On	Link up	Link up
Off	Off	Link up	Link down

How to Configure Auto-MDIX

The following sections provide configurational information about Auto-MDIX.

Configuring Auto-MDIX on an Interface

Auto MDIX is turned on by default. To disable Auto MDIX on a port, use the **no mdix auto** command under the interface configuration mode. To put it back to default, use the **mdix auto** command in the interface configuration mode. The following steps show how to enable the Auto MDIX.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode

	Command or Action	Purpose
Step 3	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet1/0/1	Specifies the physical interface to be configured, and enter interface configuration mode.
Step 4	mdix auto Example: Device(config-if)# mdix auto	Enables the Auto MDIX feature.
Step 5	end Example: Device(config-if)# end	Returns to privileged EXEC mode.
Step 6	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Example for Configuring Auto-MDIX

This example shows how to enable auto-MDIX on a port:

```
Device# configure terminal
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# mdix auto
Device(config-if)# end
```

Auto-MDIX and Operational State

Table 5: Auto-MDIX and Operational State

Auto-MDIX Setting and Operational State on an Interface	Description
Auto-MDIX on (operational: on)	Auto-MDIX is enabled and is fully functioning.
Auto-MDIX on (operational: off)	Auto-MDIX is enabled on this interface but it is not functioning. To allow auto-MDIX feature to function properly, you must also set the interface speed to be autonegotiated.
Auto-MDIX off	Auto-MDIX has been disabled with the no mdix auto command.

Additional References for Auto-MDIX

Related Documentation

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	<i>Command Reference (Catalyst 9200 Series Switches)</i>
For information about the power supplies.	<i>Cisco Catalyst 9200 Series Switches Hardware Installation Guide</i>

Feature History for Auto-MDIX

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Fuji 16.9.2	Auto-MDIX on an Interface	An automatic medium-dependent interface crossover (Auto-MDIX) enabled interface detects the required cable connection type (straight through or crossover) and configures the connection appropriately.
Cisco IOS XE Cupertino 17.9.1	Auto-MDIX on an Interface	This feature was implemented on C9200CX-12P-2X2G, C9200CX-8P-2X2G, and C9200CX-12T-2X2G models of the Cisco Catalyst 9200CX Series Switches, which were introduced in this release.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to [Cisco Feature Navigator](#).



CHAPTER 3

Configuring Ethernet Management Port

- [Prerequisites for Ethernet Management Port, on page 35](#)
- [Information About the Ethernet Management Port, on page 35](#)
- [How to Configure the Ethernet Management Port, on page 37](#)
- [Example for Configuring IP Address on Ethernet Management Interface, on page 38](#)
- [Additional References for Ethernet Management Port, on page 39](#)
- [Feature History for Ethernet Management Port, on page 39](#)

Prerequisites for Ethernet Management Port

When connecting a PC to the Ethernet management port, you must first assign an IP address.

Information About the Ethernet Management Port

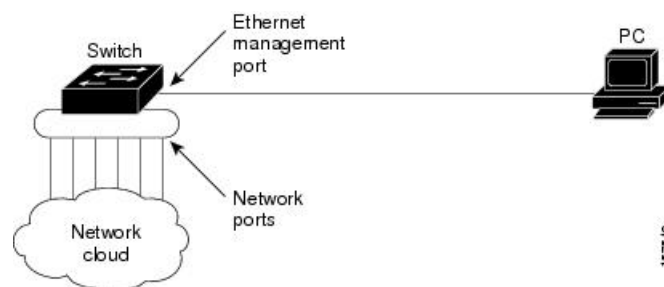
The Ethernet management port, also referred to as the *Gi0/0* or *GigabitEthernet0/0* port, is a VRF (VPN routing/forwarding) interface to which you can connect a PC. You can use the Ethernet management port instead of the device console port for network management.

When managing a device stack, connect the PC to the Ethernet management port on a stack member.

Ethernet Management Port Direct Connection to a Device

Figure 2: Connecting a Device to a PC

This figure displays how to connect the Ethernet management port to the PC for a device or a standalone device.



Ethernet Management Port Connection to Stack Devices using a Hub

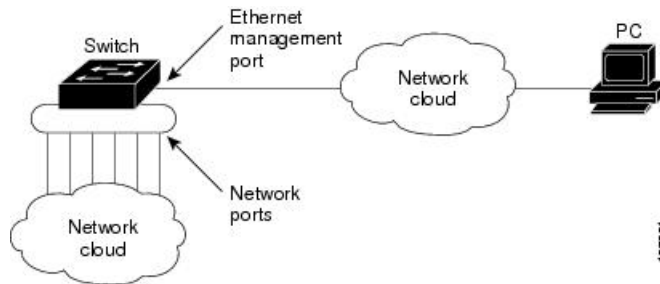
In a stack with only stack devices, all the Ethernet management ports on the stack members are connected to a hub to which the PC is connected. The active link is from the Ethernet management port on the active switch through the hub, to the PC. If the active switch fails and a new active device is elected, the active link is now from the Ethernet management port on the new active device to the PC.

Ethernet Management Port and Routing

By default, the Ethernet management port is enabled. The device cannot route packets from the Ethernet management port to a network port, and the reverse. Even though the Ethernet management port does not support routing, you may need to enable routing protocols on the port.

Figure 3: Network Example with Routing Protocols Enabled

Enable routing protocols on the Ethernet management port when the PC is multiple hops away from the device and the packets must pass through multiple Layer 3 devices to reach the PC.



In the above figure, if the Ethernet management port and the network ports are associated with the same routing process, the routes are propagated as follows:

- The routes from the Ethernet management port are propagated through the network ports to the network.
- The routes from the network ports are propagated through the Ethernet management port to the network.

Because routing is not supported between the Ethernet management port and the network ports, traffic between these ports cannot be sent or received. If this happens, data packet loops occur between the ports, which disrupt the device and network operation. To prevent the loops, configure route filters to avoid routes between the Ethernet management port and the network ports.

Supported Features on the Ethernet Management Port

The Ethernet management port supports these features:

- Express Setup (only in device stacks)
- Network Assistant
- Telnet with passwords
- TFTP
- Secure Shell (SSH)
- DHCP-based autoconfiguration

- SNMP (only ENTITY-MIB and IF-MIB)
- IP ping
- Interface features:
 - Speed: 10 Mb/s, 100 Mb/s, and autonegotiation
 - Duplex mode: Full, half, and autonegotiation
 - Loopback detection
- Cisco Discovery Protocol (CDP)
- DHCP relay agent



Caution Before enabling a feature on the Ethernet management port, make sure that the feature is supported. If you try to configure an unsupported feature on the Ethernet Management port, the feature might not work properly, and the device might fail.

How to Configure the Ethernet Management Port

This section contains the procedures of how to configure the Ethernet Management Port.

Disabling and Enabling the Ethernet Management Port

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 2	interface gigabitethernet0/0 Example: Device(config)# <code>interface gigabitethernet0/0</code>	Specifies the Ethernet management port in the CLI.
Step 3	shutdown Example: Device(config-if)# <code>shutdown</code>	Disables the Ethernet management port.
Step 4	no shutdown Example: Device(config-if)# <code>no shutdown</code>	Enables the Ethernet management port.

	Command or Action	Purpose
Step 5	exit Example: Device(config-if)# exit	Exits interface configuration mode.
Step 6	show interfaces gigabitethernet0/0 Example: Device# show interfaces gigabitethernet0/0	Displays the link status. To find out the link status to the PC, you can monitor the LED for the Ethernet management port. The LED is green (on) when the link is active, and the LED is off when the link is down. The LED is amber when there is a POST failure.

What to do next

Proceed to manage or configure your device using the Ethernet management port. See the Network Management section.

Example for Configuring IP Address on Ethernet Management Interface

This example shows how to configure IP address on the GigabitEthernet0/0 management interface.

```
Device# configure terminal
Device(config)# interface gigabitethernet0/0
Device(config-if)# vrf forwarding Mgmt-vrf
Device(config-if)# ip address 192.168.247.10 255.255.0.0
Device(config-if)# end
```

```
Device# show running-config interface Gi0/0
```

```
Building configuration...
```

```
Current configuration : 118 bytes
!
interface GigabitEthernet0/0
 vrf forwarding Mgmt-vrf
 ip address 192.168.247.10 255.255.0.0
 negotiation auto
end
```

This example shows how to configure IP address on the TenGigabitEthernet0/1 management interface.

```
Device# configure terminal
Device(config)# interface TenGigabitEthernet0/1
Device(config-if)# vrf forwarding Mgmt-vrf
Device(config-if)# ip address 192.168.247.20 255.255.0.0
Device(config-if)# negotiation auto
Device(config-if)# end
```

```
Device# show running-config interface Te0/1
```

```
Building configuration...
```



```

Current configuration : 118 bytes
!
interface TenGigabitEthernet0/1
 vrf forwarding Mgmt-vrf
 ip address 192.168.247.20 255.255.0.0
 negotiation auto
end

```

Additional References for Ethernet Management Port

Related Documents

Related Topic	Document Title
Bootloader configuration	See the <i>System Management</i> section of this guide.
Bootloader commands	See the <i>System Management Commands</i> section of the <i>Command Reference (Catalyst 9200 Series Switches)</i> .

Feature History for Ethernet Management Port

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Fuji 16.9.2	Ethernet Management Port	The Ethernet management port is a VRF interface to which you can connect a PC. You can use the Ethernet management port instead of the device console port for network management.
Cisco IOS XE Cupertino 17.9.1	Ethernet Management Port	This feature was implemented on C9200CX-12P-2X2G, C9200CX-8P-2X2G, and C9200CX-12T-2X2G models of the Cisco Catalyst 9200CX Series Switches, which were introduced in this release.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to [Cisco Feature Navigator](#).



CHAPTER 4

Checking Port Status and Connectivity

- [Check Cable Status Using Time Domain Reflectometer, on page 41](#)
- [Feature History for Checking Port Status and Connectivity, on page 42](#)

Check Cable Status Using Time Domain Reflectometer

The Time Domain Reflectometer (TDR) feature allows you to determine if a cable is OPEN or SHORT when it is at fault.

Running the TDR Test

To start the TDR test, perform this task:

Procedure

	Command or Action	Purpose
Step 1	test cable-diagnostics tdr {interface { interface-number }}	Starts the TDR test.
Step 2	show cable-diagnostics tdr {interface interface-number}	Displays the TDR test counter information.

TDR Guidelines

The following guidelines apply to the use of TDR:

- Do not change the port configuration while the TDR test is running.
- If you connect a port undergoing a TDR test to an Auto-MDIX enabled port, the TDR result might be invalid.
- If you connect a port undergoing a TDR test to a 100BASE-T port such as that on the device, the unused pairs (4-5 and 7-8) are reported as faulty because the remote end does not terminate these pairs.
- To run a TDR test, the cable length should be at least 10 meters. If the cable is shorter than 10 meters, the test is considered as invalid.

- Due to cable characteristics, you should run the TDR test multiple times to get accurate results.
- Do not change port status (for example, remove the cable at the near or far end) because the results might be inaccurate.
- TDR works best if the test cable is disconnected from the remote port. Otherwise, it might be difficult for you to interpret results correctly.
- TDR operates across four wires. Depending on the cable conditions, the status might show that one pair is OPEN or SHORT while all other wire pairs display as faulty. This operation is acceptable because you should declare a cable faulty provided one pair of wires is either OPEN or SHORT.
- TDR intent is to determine how poorly a cable is functioning rather than to locate a faulty cable.
- When TDR locates a faulty cable, you should still use an offline cable diagnosis tool to better diagnose the problem.

Feature History for Checking Port Status and Connectivity

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.12.1	Time Domain Reflectometer (TDR)	TDR allows you to determine if a cable is OPEN or SHORT when it is at fault.
Cisco IOS XE Cupertino 17.9.1	Port Status and Connectivity Check	This feature was implemented on C9200CX-12P-2X2G, C9200CX-8P-2X2G, and C9200CX-12T-2X2G models of the Cisco Catalyst 9200CX Series Switches, which were introduced in this release.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to [Cisco Feature Navigator](#).



CHAPTER 5

Configuring LLDP, LLDP-MED, and Wired Location Service

- [Restrictions for LLDP, on page 43](#)
- [Information About LLDP, LLDP-MED, and Wired Location Service, on page 43](#)
- [How to Configure LLDP, LLDP-MED, and Wired Location Service, on page 47](#)
- [Configuration Examples for LLDP, LLDP-MED, and Wired Location Service, on page 55](#)
- [Monitoring and Maintaining LLDP, LLDP-MED, and Wired Location Service, on page 56](#)
- [Additional References for LLDP, LLDP-MED, and Wired Location Service, on page 57](#)
- [Feature History for LLDP, LLDP-MED, and Wired Location Service, on page 57](#)

Restrictions for LLDP

- If the interface is configured as a tunnel port, LLDP is automatically disabled.
- If you first configure a network-policy profile on an interface, you cannot apply the **switchport voice vlan** command on the interface. If the **switchport voice vlan *vlan-id*** is already configured on an interface, you can apply a network-policy profile on the interface. This way the interface has the voice or voice-signaling VLAN network-policy profile applied on the interface.
- You cannot configure static secure MAC addresses on an interface that has a network-policy profile.
- When Cisco Discovery Protocol and LLDP are both in use within the same switch, it is necessary to disable LLDP on interfaces where Cisco Discovery Protocol is in use for power negotiation. LLDP can be disabled at interface level with the commands **no lldp tlv-select power-management** or **no lldp transmit / no lldp receive**.

Information About LLDP, LLDP-MED, and Wired Location Service

This section describes about LLDP, LLDP-MED, and wired location service.

LLDP

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, switches, and controllers). CDP allows

network management applications to automatically discover and learn about other Cisco devices connected to the network.

To support non-Cisco devices and to allow for interoperability between other devices, the device supports the IEEE 802.1AB Link Layer Discovery Protocol (LLDP). LLDP is a neighbor discovery protocol that is used for network devices to advertise information about themselves to other devices on the network. This protocol runs over the data-link layer, which allows two systems running different network layer protocols to learn about each other.

LLDP Supported TLVs

LLDP supports a set of attributes that it uses to discover neighbor devices. These attributes contain type, length, and value descriptions and are referred to as TLVs. LLDP supported devices can use TLVs to receive and send information to their neighbors. This protocol can advertise details such as configuration information, device capabilities, and device identity.

The switch supports these basic management TLVs. These are mandatory LLDP TLVs.

- Port description TLV
- System name TLV
- System description TLV
- System capabilities TLV
- Management address TLV

These organizationally specific LLDP TLVs are also advertised to support LLDP-MED.

- Port VLAN ID TLV (IEEE 802.1 organizationally specific TLVs)
- MAC/PHY configuration/status TLV (IEEE 802.3 organizationally specific TLVs)

LLDP-MED

LLDP for Media Endpoint Devices (LLDP-MED) is an extension to LLDP that operates between endpoint devices such as IP phones and network devices. It specifically provides support for voice over IP (VoIP) applications and provides additional TLVs for capabilities discovery, network policy, Power over Ethernet, inventory management and location information. By default, all LLDP-MED TLVs are enabled.

LLDP-MED Supported TLVs

LLDP-MED supports these TLVs:

- LLDP-MED capabilities TLV

Allows LLDP-MED endpoints to determine the capabilities that the connected device supports and has enabled.

- Network policy TLV

Allows both network connectivity devices and endpoints to advertise VLAN configurations and associated Layer 2 and Layer 3 attributes for the specific application on that port. For example, the switch can notify a phone of the VLAN number that it should use. The phone can connect to any device, obtain its VLAN number, and then start communicating with the call control.

By defining a network-policy profile TLV, you can create a profile for voice and voice-signaling by specifying the values for VLAN, class of service (CoS), differentiated services code point (DSCP), and tagging mode. These profile attributes are then maintained centrally on the switch and propagated to the phone.

- Power management TLV

Enables advanced power management between LLDP-MED endpoint and network connectivity devices. Allows devices and phones to convey power information, such as how the device is powered, power priority, and how much power the device needs.

LLDP-MED also supports an extended power TLV to advertise fine-grained power requirements, end-point power priority, and end-point and network connectivity-device power status. LLDP is enabled and power is applied to a port, the power TLV determines the actual power requirement of the endpoint device so that the system power budget can be adjusted accordingly. The device processes the requests and either grants or denies power based on the current power budget. If the request is granted, the switch updates the power budget. If the request is denied, the device turns off power to the port, generates a syslog message, and updates the power budget. If LLDP-MED is disabled or if the endpoint does not support the LLDP-MED power TLV, the initial allocation value is used throughout the duration of the connection.

You can change power settings by entering the **power inline** {**auto** [**max** *max-wattage*] | **never** | **static** [**max** *max-wattage*] } interface configuration command. By default the PoE interface is in **auto** mode;

- Inventory management TLV

Allows an endpoint to send detailed inventory information about itself to the device, including information hardware revision, firmware version, software version, serial number, manufacturer name, model name, and asset ID TLV.

- Location TLV

Provides location information from the device to the endpoint device. The location TLV can send this information:

- Civic location information

Provides the civic address information and postal information. Examples of civic location information are street address, road name, and postal community name information.

- ELIN location information

Provides the location information of a caller. The location is determined by the Emergency location identifier number (ELIN), which is a phone number that routes an emergency call to the local public safety answering point (PSAP) and which the PSAP can use to call back the emergency caller.

- Geographic location information

Provides the geographical details of a switch location such as latitude, longitude, and altitude of a switch.

- custom location

Provides customized name and value of a switch location.

Wired Location Service

The device uses the location service feature to send location and attachment tracking information for its connected devices to a Cisco Mobility Services Engine (MSE). The tracked device can be a wireless endpoint, a wired endpoint, or a wired device or controller. The device notifies the MSE of device link up and link down events through the Network Mobility Services Protocol (NMSP) location and attachment notifications.

The MSE starts the NMSP connection to the device, which opens a server port. When the MSE connects to the device there are a set of message exchanges to establish version compatibility and service exchange information followed by location information synchronization. After connection, the device periodically sends location and attachment notifications to the MSE. Any link up or link down events detected during an interval are aggregated and sent at the end of the interval.

When the device determines the presence or absence of a device on a link-up or link-down event, it obtains the client-specific information such as the MAC address, IP address, and username. If the client is LLDP-MED- or CDP-capable, the device obtains the serial number and UDI through the LLDP-MED location TLV or CDP.

Depending on the device capabilities, the device obtains this client information at link up:

- Slot and port specified in port connection
- MAC address specified in the client MAC address
- IP address specified in port connection
- 802.1X username if applicable
- Device category is specified as a *wired station*
- State is specified as *new*
- Serial number, UDI
- Model number
- Time in seconds since the device detected the association

Depending on the device capabilities, the device obtains this client information at link down:

- Slot and port that was disconnected
- MAC address
- IP address
- 802.1X username if applicable
- Device category is specified as a *wired station*
- State is specified as *delete*
- Serial number, UDI
- Time in seconds since the device detected the disassociation

When the device shuts down, it sends an attachment notification with the state *delete* and the IP address before closing the NMSP connection to the MSE. The MSE interprets this notification as disassociation for all the wired clients associated with the device.

If you change a location address on the device, the device sends an NMSP location notification message that identifies the affected ports and the changed address information.

Default LLDP Configuration

Table 6: Default LLDP Configuration

Feature	Default Setting
LLDP global state	Disabled
LLDP holdtime (before discarding)	120 seconds
LLDP timer (packet update frequency)	30 seconds
LLDP reinitialization delay	2 seconds
LLDP tlv-select	Disabled to send and receive all TLVs
LLDP interface state	Disabled
LLDP receive	Disabled
LLDP transmit	Disabled
LLDP med-tlv-select	Disabled to send all LLDP-MED TLVs. When LLDP is glob LLDP-MED-TLV is also enabled.

How to Configure LLDP, LLDP-MED, and Wired Location Service

This section provides the procedures to configure LLDP, LLDP-MED, and wired location service.

Enabling LLDP

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	lldp run Example:	Enables LLDP globally on the device.

	Command or Action	Purpose
	Device(config)# lldp run	
Step 4	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet2/0/1	Specifies the interface on which you are enabling LLDP, and enters interface configuration mode.
Step 5	lldp transmit Example: Device(config-if)# lldp transmit	Enables the interface to send LLDP packets.
Step 6	lldp receive Example: Device(config-if)# lldp receive	Enables the interface to receive LLDP packets.
Step 7	end Example: Device(config-if)# end	Exits interface configuration mode, and returns to privileged EXEC mode.
Step 8	show lldp Example: Device# show lldp	Verifies the configuration.
Step 9	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring LLDP Characteristics

You can configure the frequency of LLDP updates, the amount of time to hold the information before discarding it, and the initialization delay time. You can also select the LLDP and LLDP-MED TLVs to send and receive.



Note Steps 3 through 6 are optional and can be performed in any order.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	lldp holdtime seconds Example: Device (config)# <code>lldp holdtime 120</code>	(Optional) Specifies the amount of time a receiving device should hold the information from your device before discarding it. The range is 0 to 65535 seconds; the default is 120 seconds.
Step 4	lldp reinit delay Example: Device (config)# <code>lldp reinit 2</code>	(Optional) Specifies the delay time in seconds for LLDP to initialize on an interface. The range is 2 to 5 seconds; the default is 2 seconds.
Step 5	lldp timer rate Example: Device (config)# <code>lldp timer 30</code>	(Optional) Sets the sending frequency of LLDP updates in seconds. The range is 5 to 65534 seconds; the default is 30 seconds.
Step 6	lldp tlv-select Example: Device (config)# <code>tlv-select</code>	(Optional) Specifies the LLDP TLVs to send or receive.
Step 7	interface interface-id Example: Device (config)# <code>interface gigabitethernet2/0/1</code>	Specifies the interface on which you are enabling LLDP, and enters interface configuration mode.
Step 8	lldp med-tlv-select Example: Device (config-if)# <code>lldp med-tlv-select inventory management</code>	(Optional) Specifies the LLDP-MED TLVs to send or receive.
Step 9	end Example: Device (config-if)# <code>end</code>	Exits interface configuration mode, and returns to privileged EXEC mode.
Step 10	show lldp Example: Device# <code>show lldp</code>	Verifies the configuration.
Step 11	copy running-config startup-config Example: Device# <code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Configuring LLDP-MED TLVs

By default, the device only sends LLDP packets until it receives LLDP-MED packets from the end device. It then sends LLDP packets with MED TLVs, as well. When the LLDP-MED entry has been aged out, it again only sends LLDP packets.

By using the **lldp** interface configuration command, you can configure the interface not to send the TLVs listed in the following table.

Table 7: LLDP-MED TLVs

LLDP-MED TLV	Description
inventory-management	LLDP-MED inventory management TLV
location	LLDP-MED location TLV
network-policy	LLDP-MED network policy TLV
power-management	LLDP-MED power management TLV

Follow these steps to enable a TLV on an interface:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface interface-id Example: Device(config)# interface gigabitethernet2/0/1	Specifies the interface on which you are enabling LLDP, and enters interface configuration mode.
Step 4	lldp med-tlv-select Example: Device(config-if)# lldp med-tlv-select inventory management	Specifies the TLV to enable.
Step 5	end Example: Device(config-if)# end	Exits global configuration mode, and returns to privileged EXEC mode.
Step 6	copy running-config startup-config Example:	(Optional) Saves your entries in the configuration file.

	Command or Action	Purpose
	Device# <code>copy running-config startup-config</code>	

Configuring Network-Policy TLV

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> <code>enable</code>	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# <code>configure terminal</code>	Enters global configuration mode.
Step 3	network-policy profile <i>profile number</i> Example: Device (config)# <code>network-policy profile 1</code>	Specifies the network-policy profile number, and enters network-policy configuration mode. The range is 1 to 4294967295.
Step 4	{voice voice-signaling} vlan [<i>vlan-id</i> { cos <i>cvalue</i> dscp <i>dvalue</i> }] [[dot1p { cos <i>cvalue</i> dscp <i>dvalue</i> }] none untagged] Example: Device (config-network-policy)# <code>voice vlan 100 cos 4</code>	Configures the policy attributes: <ul style="list-style-type: none"> • voice: Specifies the voice application type. • voice-signaling: Specifies the voice-signaling application type. • vlan: Specifies the native VLAN for voice traffic. • <i>vlan-id</i>: (Optional) Specifies the VLAN for voice traffic. The range is 1 to 4094. • cos cvalue: (Optional) Specifies the Layer 2 priority class of service (CoS) for the configured VLAN. The range is 0 to 7; the default is 5. • dscp dvalue: (Optional) Specifies the differentiated services code point (DSCP) value for the configured VLAN. The range is 0 to 63; the default is 46. • dot1p: (Optional) Configures the telephone to use IEEE 802.1p priority tagging and use VLAN 0 (the native VLAN).

	Command or Action	Purpose
		<ul style="list-style-type: none"> • none: (Optional) Do not instruct the IP telephone about the voice VLAN. The telephone uses the configuration from the telephone key pad. • untagged: (Optional) Configures the telephone to send untagged voice traffic. This is the default for the telephone.
Step 5	exit Example: Device (config) # exit	Returns to global configuration mode.
Step 6	interface <i>interface-id</i> Example: Device (config) # interface gigabitethernet2/0/1	Specifies the interface on which you are configuring a network-policy profile, and enters interface configuration mode.
Step 7	network-policy <i>profile number</i> Example: Device (config-if) # network-policy 1	Specifies the network-policy profile number.
Step 8	lldp med-tlv-select network-policy Example: Device (config-if) # lldp med-tlv-select network-policy	Specifies the network-policy TLV.
Step 9	end Example: Device (config) # end	Returns to privileged EXEC mode.
Step 10	show network-policy profile Example: Device# show network-policy profile	Verifies the configuration.
Step 11	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring Location TLV and Wired Location Service

Beginning in privileged EXEC mode, follow these steps to configure location information for an endpoint and to apply it to an interface.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	location { admin-tag <i>string</i> civic-location identifier { <i>id</i> <i>host</i> } elin-location string identifier <i>id</i> custom-location identifier { <i>id</i> <i>host</i> } geo-location identifier { <i>id</i> <i>host</i> }} Example: Device(config)# location civic-location identifier 1 Device(config-civic)# number 3550 Device(config-civic)# primary-road-name "Cisco Way" Device(config-civic)# city "San Jose" Device(config-civic)# state CA Device(config-civic)# building 19 Device(config-civic)# room C6 Device(config-civic)# county "Santa Clara" Device(config-civic)# country US	Specifies the location information for an endpoint. <ul style="list-style-type: none"> • admin-tag: Specifies an administrative tag or site information. • civic-location: Specifies civic location information. • elin-location: Specifies emergency location information (ELIN). • custom-location: Specifies custom location information. • geo-location: Specifies geo-spatial location information. • identifier id: Specifies the ID for the civic, ELIN, custom, or geo location. • host: Specifies the host civic, custom, or geo location. • string: Specifies the site or location information in alphanumeric format.
Step 3	exit Example: Device(config-civic)# exit	Returns to global configuration mode.
Step 4	interface <i>interface-id</i> Example:	Specifies the interface on which you are configuring the location information, and enters interface configuration mode.
Step 5	location { additional-location-information <i>word</i> civic-location-id { <i>id</i> <i>host</i> } elin-location-id <i>id</i> custom-location-id { <i>id</i> <i>host</i> } geo-location-id { <i>id</i> <i>host</i> }} Example: Device(config-if)# location elin-location-id 1	Enters location information for an interface: <ul style="list-style-type: none"> • additional-location-information: Specifies additional information for a location or place. • civic-location-id: Specifies global civic location information for an interface. • elin-location-id: Specifies emergency location information for an interface.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • custom-location-id: Specifies custom location information for an interface. • geo-location-id: Specifies geo-spatial location information for an interface. • host: Specifies the host location identifier. • word: Specifies a word or phrase with additional location information. • id: Specifies the ID for the civic, ELIN, custom, or geo location. The ID range is 1 to 4095.
Step 6	end Example: Device(config-if)# end	Exits interface configuration mode, and returns to privileged EXEC mode.
Step 7	Use one of the following: <ul style="list-style-type: none"> • show location admin-tag <i>string</i> • show location civic-location identifier <i>id</i> • show location elin-location identifier <i>id</i> Example: Device# show location admin-tag OR Device# show location civic-location identifier OR Device# show location elin-location identifier	Verifies the configuration.
Step 8	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Enabling Wired Location Service on the Device

Before you begin

For wired location to function, you must first enter the **ip device tracking** global configuration command.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	nmosp notification interval {attachment location} interval-seconds Example: Device(config)# nmosp notification interval location 10	Specifies the NMSP notification interval. attachment: Specifies the attachment notification interval. location: Specifies the location notification interval. <i>interval-seconds:</i> Duration in seconds before the device sends the MSE the location or attachment updates. The range is 1 to 30; the default is 30.
Step 4	end Example: Device(config)# end	Exits global configuration mode, and returns to privileged EXEC mode.
Step 5	show network-policy profile Example: Device# show network-policy profile	Verifies the configuration.
Step 6	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuration Examples for LLDP, LLDP-MED, and Wired Location Service

This section provides configuration examples for LLDP, LLDP-MED, and wired location service.

Examples: Configuring Network-Policy TLV

This example shows how to configure VLAN 100 for voice application with CoS and to enable the network-policy profile and network-policy TLV on an interface:

```

Device# configure terminal
Device(config)# network-policy 1
Device(config-network-policy)# voice vlan 100 cos 4
Device(config-network-policy)# exit
Device(config)# interface gigabitethernet1/0/1
Device(config-if)# network-policy profile 1
Device(config-if)# lldp med-tlv-select network-policy

```

This example shows how to configure the voice application type for the native VLAN with priority tagging:

```

Device-config-network-policy)# voice vlan dot1p cos 4
Device-config-network-policy)# voice vlan dot1p dscp 34

```

Monitoring and Maintaining LLDP, LLDP-MED, and Wired Location Service

Use the following commands for monitoring and maintaining LLDP, LLDP-MED, and wired location service.

Command	Description
clear lldp counters	Resets the traffic counters to zero.
clear lldp table	Deletes the LLDP neighbor information table.
clear nmosp statistics	Clears the NMSP statistic counters.
show lldp	Displays global information, such as frequency of transmissions, the holdtime for packets being sent, and the delay time before LLDP initializes on an interface.
show lldp entry <i>entry-name</i>	Displays information about a specific neighbor. You can enter an asterisk (*) to display all neighbors, or you can enter the neighbor name.
show lldp interface [<i>interface-id</i>]	Displays information about interfaces with LLDP enabled. You can limit the display to a specific interface.
show lldp neighbors [<i>interface-id</i>] [detail]	Displays information about neighbors, including device type, interface type and number, holdtime settings, capabilities, and port ID. You can limit the display to neighbors of a specific interface or expand the display for more detailed information.
show lldp traffic	Displays LLDP counters, including the number of packets sent and received, number of packets discarded, and number of unrecognized TLVs.

Command	Description
<code>show location admin-tag <i>string</i></code>	Displays the location information for the specified administrative tag or site.
<code>show location civic-location identifier <i>id</i></code>	Displays the location information for a specific global civic location.
<code>show location elin-location identifier <i>id</i></code>	Displays the location information for an emergency location
<code>show network-policy profile</code>	Displays the configured network-policy profiles.
<code>show nmsp</code>	Displays the NMSp information

Additional References for LLDP, LLDP-MED, and Wired Location Service

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the <i>Interface and Hardware Commands</i> section in the <i>Command Reference (Catalyst 9200 Series Switches)</i>

Feature History for LLDP, LLDP-MED, and Wired Location Service

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Fuji 16.9.2	Link Layer Discovery Protocol (LLDP), LLDP-MED, Wired Location Service	<p>LLDP is a neighbor discovery protocol that is used for network devices to advertise information about themselves to other devices on the network. This protocol runs over the data-link layer, which allows two systems running different network layer protocols to learn about each other.</p> <p>LLDP-MED operates between endpoints and network devices.</p> <p>Wired Location Service lets you send tracking information of the connected devices to a Cisco Mobility Services Engine (MSE).</p>

Release	Feature	Feature Information
Cisco IOS XE Cupertino 17.9.1	Link Layer Discovery Protocol (LLDP), LLDP-MED, Wired Location Service	This feature was implemented on C9200CX-12P-2X2G, C9200CX-8P-2X2G, and C9200CX-12T-2X2G models of the Cisco Catalyst 9200CX Series Switches, which were introduced in this release.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to [Cisco Feature Navigator](#).



CHAPTER 6

Configuring System MTU

- [Restrictions for System MTU, on page 59](#)
- [Information About the MTU, on page 59](#)
- [How to Configure MTU , on page 59](#)
- [Configuration Examples for System MTU, on page 61](#)
- [Additional References for System MTU, on page 62](#)
- [Feature History for System MTU, on page 62](#)

Restrictions for System MTU

- Do not configure the system MTU value while the traffic is flowing.

Information About the MTU

The default maximum transmission unit (MTU) size for payload received in Ethernet frame and sent on all device interfaces is 1500 bytes.

System MTU Value Application

The upper limit of the IP or IPv6 MTU value is based on the switch or switch stack configuration and refers to the currently applied system MTU value. For more information about setting the MTU sizes, see the **system mtu** global configuration command in the command reference for this release.

Beginning from Cisco IOS XE Amsterdam 17.3.x, the minimum IPv6 system MTU is fixed at 1280 as per RFC 8200.

How to Configure MTU

The following tasks describe how you can configure MTU.

Configuring the System MTU

Follow these steps to change the MTU size for switched packets:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	system mtu bytes Example: Device(config)# system mtu 1900	(Optional) Changes the MTU size for all interfaces.
Step 4	end Example: Device(config)# end	Enters global configuration mode, and returns to privileged EXEC mode.
Step 5	copy running-config startup-config Example: Device# copy running-config startup-config	Saves your entries in the configuration file.
Step 6	show system mtu Example: Device# show system mtu	Verifies your settings.

Configuring Protocol-Specific MTU

To override system MTU values on routed interfaces, configure protocol-specific MTU under each routed interface. To change the MTU size for routed ports, perform this procedure.

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 2	interface <i>interface</i> Example: Device(config)# interface gigabitethernet0/0	Enters interface configuration mode.
Step 3	ip mtu <i>bytes</i> Example: Device(config-if)# ip mtu 68	Changes the IPv4 MTU size
Step 4	ipv6 mtu <i>bytes</i> Example: Device(config-if)# ipv6 mtu 1280	(Optional) Changes the IPv6 MTU size.
Step 5	end Example: Device(config-if)# end	Exits interface configuration mode, and returns to privileged EXEC mode.
Step 6	copy running-config startup-config Example: Device# copy running-config startup-config	Saves your entries in the configuration file.
Step 7	show system mtu Example: Device# show system mtu	Verifies your settings.

Configuration Examples for System MTU

Example: Configuring Protocol-Specific MTU

This example shows how you can configure protocol-specific MTU:

```
Device# configure terminal
Device(config)# interface gigabitethernet 0/1
Device(config-if)# ip mtu 900
Device(config-if)# ipv6 mtu 1286
Device(config-if)# end
```

Example: Configuring the System MTU

This example shows how you can configure the system MTU:

```
Device# configure terminal
Device(config)# system mtu 1600
Device(config)# exit
```

Additional References for System MTU

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the <i>Interface and Hardware Commands</i> section in the <i>Command Reference (Catalyst 9200 Series Switches)</i>

Standards and RFCs

Standard/RFC	Title
RFC 8200	<i>Internet Protocol, Version 6 (IPv6) Specification</i>

Feature History for System MTU

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Fuji 16.9.2	System MTU	System MTU defines the maximum transmission unit size for frames transmitted on all interfaces of a switch.
Cisco IOS XE Cupertino 17.9.1	System MTU	This feature was implemented on C9200CX-12P-2X2G, C9200CX-8P-2X2G, and C9200CX-12T-2X2G models of the Cisco Catalyst 9200CX Series Switches, which were introduced in this release.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to [Cisco Feature Navigator](#).



CHAPTER 7

Configuring Per-Port MTU

- [Restrictions for Per-Port MTU, on page 63](#)
- [Information About Per-Port MTU, on page 63](#)
- [Configuring Per-Port MTU, on page 64](#)
- [Example: Configuring Per-Port MTU, on page 64](#)
- [Example: Verifying Per-Port MTU, on page 65](#)
- [Example: Disabling Per-Port MTU, on page 65](#)
- [Feature History for Per-Port MTU, on page 65](#)

Restrictions for Per-Port MTU

- Per-Port MTU cannot be configured on the management port.
- Per-Port MTU cannot be configured on SVL links.
- Members of a port channel cannot be configured with Per-Port MTU, they derive their MTU from the port-channel MTU configuration.
- Per-Port MTU is not supported on subinterfaces and port-channel subinterfaces.
- Do not configure the per-port MTU value while the traffic is flowing.

Information About Per-Port MTU

You can configure the MTU size for all interfaces on a device at the same time using the **system mtu** command. The default maximum transmission unit (MTU) size for frames received and transmitted on all interfaces is 1500 bytes. The **system mtu** command is a global command and does not allow MTU to be configured at a port level. Starting with Cisco IOS XE 17.1.1, you can configure Per-Port MTU. Per-Port MTU will support port level and port channel level MTU configuration. With Per-Port MTU you can set different MTU values for different interfaces as well as different port channel interfaces.

Per-port MTU can be configured in the range of 1500-9198 bytes.

Once the Per-Port MTU value has been configured on a port, the protocol-specific MTU for that port is also changed to the Per-Port MTU value. When Per-Port MTU is configured on a port, you can still configure protocol-specific MTU on the interface in the range from 256 to Per-Port MTU value.

If the Per-Port MTU is disabled, the MTU for the port will revert to the system MTU value.

You can view the Per-Port MTU configurations on an interface using the **show interface mtu** command.

The following are expected behaviour if the Per-Port MTU configuration is changed on any interface:

- The interface flaps if the port-channel is in PAgP or LACP mode.
- The interface does not flap if the port channel is in the **on** mode.
- The interface does not flap if the interface is not a port channel.

You can disable Per-Port MTU by using the **no** form of the **mtu bytes** command in the interface configuration mode.

Configuring Per-Port MTU

Follow these steps to change the MTU size for switched packets on a particular port of an interface:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>typeswitch-number/slot-number/port-number</i> Example: Device(config)# int FortyGigabitEthernet2/5/0/20	Configures the interface and enters interface configuration mode.
Step 4	mtu bytes Example: Device(config-if)# mtu 6666	Configures the MTU size for a particular port on the interface.
Step 5	end Example: Device(config-if)# end	Exits interface configuration mode, and returns to privileged EXEC mode.

Example: Configuring Per-Port MTU

This example shows how to configure Per-Port MTU on an interface:

```
Device# configure terminal
Device(config)# interface FortyGigabitEthernet2/5/0/20
```

```
Device(config-if)# mtu 6666
Device(config-if)# end
```

Example: Verifying Per-Port MTU

This example shows how to verify Per-Port MTU on an interface using the **show interface mtu** command:

```
Device# show interface mtu

Port          Name          MTU
Fo2/5/0/19   Name          1500
Fo2/5/0/20   Name          6666
Fo2/5/0/21   ixia_7_21    1500
```

Example: Disabling Per-Port MTU

This example shows how to disable Per-Port MTU on an interface:

```
Device# configure terminal
Device(config)# interface FortyGigabitEthernet2/5/0/20
Device(config-if)# no mtu
Device(config-if)# end
```

Feature History for Per-Port MTU

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Amsterdam 17.1.1	Per-Port MTU	Per-Port MTU defines the maximum transmission unit size for frames received and transmitted on a particular port or port channel.
Cisco IOS XE Cupertino 17.9.1	Per-Port MTU	This feature was implemented on C9200CX-12P-2X2G, C9200CX-8P-2X2G, and C9200CX-12T-2X2G models of the Cisco Catalyst 9200CX Series Switches, which were introduced in this release.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to [Cisco Feature Navigator](#).



CHAPTER 8

Configuring Internal Power Supplies

- [Information About Internal Power Supplies](#), on page 67
- [How to Configure Internal Power Supplies](#), on page 67
- [Monitoring Internal Power Supplies](#), on page 68
- [Configuration Examples for Internal Power Supplies](#), on page 68
- [Additional References for Internal Power Supplies](#), on page 69
- [Feature History for Internal Power Supplies](#), on page 69

Information About Internal Power Supplies

See the device installation guide for information about the power supplies.

How to Configure Internal Power Supplies

This section list the procedures to configure internal power supplies.

Configuring Internal Power Supply

You can use the **power supply** EXEC command to configure and manage the internal power supply on the device. The device does not support the **no power supply** EXEC command.

Follow these steps beginning in user EXEC mode:

Procedure

	Command or Action	Purpose
Step 1	<p>power supply <i>switch_number</i> slot{A B} { off on }</p> <p>Example: Device# power supply 1 slot A on</p>	<p>Sets the specified power supply to off or on by using one of these keywords:</p> <ul style="list-style-type: none"> • A: Selects the power supply in slot A. • B: Selects power supply in slot B. <p>Note Power supply slot B is the closest to the outer edge of the device.</p>

	Command or Action	Purpose
		<ul style="list-style-type: none"> • off: Sets the power supply off. • on: Sets the power supply on. <p>By default, the device power supply is on.</p>
Step 2	show environment power Example: Device# <code>show environment power</code>	Verifies your settings.

Monitoring Internal Power Supplies

Table 8: Show Commands for Power Supplies

Command	Purpose
<code>show environment power [all switch <i>switch_number</i>]</code>	(Optional) Displays the status of the internal power supplies for each device in the stack or for the specified device. The range is 1 to 8, depending on the device member numbers in the stack. The device keywords are available only on stacking-capable devices.

Configuration Examples for Internal Power Supplies

This example shows a sample output of the `show power detail` command when a power supply with serial number starting with QCS is installed on the C9500X-60L4D switch.

Device# `show power detail`

```

Power
Supply  Model No          Type  Capacity  Fan States
-----  -----  ---  -
                Status    0      1
-----  -----  ---  -
PS1     C9K-PWR-1500WAC     AC    1500 W    active   good  n.a
PS2     C9K-PWR-1500WAC     AC    1500 W    active   good  n.a

```

```

Fan
Tray    Status    Fan states
-----  -----  -
                0      1
-----  -----  -
FT1     active   good  good
FT2     active   good  good
FT3     active   good  good
FT4     active   good  good
FT5     active   good  good
FT6     active   good  good

```

This example shows how to set the power supply in slot A to off:

```
Device# power supply 1 slot A off
Disabling Power supply A may result in a power loss to PoE devices and/or switches
Device#
Jun 10 04:52:54.389: %PLATFORM_ENV-6-FRU_PS_OIR: FRU Power Supply 1 powered off
Jun 10 04:52:56.717: %PLATFORM_ENV-1-FAN_NOT_PRESENT: Fan is not present
Device#
```

This example shows how to set the power supply in slot A to on:

```
Device# power supply 1 slot A on
Jun 10 04:54:39.600: %PLATFORM_ENV-6-FRU_PS_OIR: FRU Power Supply 1 powered on
```

This example shows the output of the **show env power** command:

Table 9: show env power Status Descriptions

Field	Description
OK	The power supply is present and power is good.
Not Present	No power supply is installed.
No Input Power	The power supply is present but there is no input power.
Disabled	The power supply and input power are present, but power supply is switched off by CLI.
No Response	The power supply is not recognizable or is faulty.
Failure-Fan	The power supply fan is faulty.

Additional References for Internal Power Supplies

Related Documentation

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	<i>Command Reference (Catalyst 9200 Series Switches)</i>
For information about the power supplies.	<i>Cisco Catalyst 9200 Series Switches Hardware Installation Guide</i>
For information about PoE Port Priority and Load Shedding.	<i>Configuring Interface Characteristics Chapter in Interface and Hardware Components Configuration Guide (Catalyst 9200 Series Switches).</i>

Feature History for Internal Power Supplies

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Fuji 16.9.2	Internal Power Supplies	The switch operates with power supply modules which could be AC, DC or both. Refer the <i>Hardware Installation Guide</i> for more details on power supply units.
Cisco IOS XE Cupertino 17.9.1	Internal Power Supplies	This feature was implemented on C9200CX-12P-2X2G, C9200CX-8P-2X2G, and C9200CX-12T-2X2G models of the Cisco Catalyst 9200CX Series Switches, which were introduced in this release.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to [Cisco Feature Navigator](#).



CHAPTER 9

Configuring EEE

- [Restrictions for EEE, on page 71](#)
- [Information About EEE, on page 71](#)
- [How to Configure EEE, on page 72](#)
- [Monitoring EEE, on page 73](#)
- [Configuration Examples for Configuring EEE, on page 73](#)
- [Additional References for EEE, on page 74](#)
- [Feature History for Configuring EEE, on page 74](#)

Restrictions for EEE

Energy Efficient Ethernet (EEE) has the following restrictions:

- Changing the EEE configuration resets the interface because the device has to restart Layer 1 autonegotiation.
- We recommend that you enable Link Layer Discovery Protocol (LLDP) for devices that require longer wakeup times before they are able to accept data on their receive paths. This enables the device to negotiate for extended system wakeup times from the transmitting link partner.

Information About EEE

This section provides information about EEE.

EEE Overview

Energy Efficient Ethernet (EEE) is an IEEE 802.3az standard that is designed to reduce power consumption in Ethernet networks during idle periods.

Default EEE Configuration

EEE is disabled by default.

How to Configure EEE

You can enable or disable EEE on an interface that is connected to an EEE-capable link partner.

Enabling or Disabling EEE

Procedure

	Command or Action	Purpose
Step 1	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 2	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet 1/0/1	Specifies the interface to be configured, and enters interface configuration mode.
Step 3	power efficient-ethernet auto Example: Device(config-if)# power efficient-ethernet auto	Enables EEE on the specified interface. When EEE is enabled, the device advertises and autonegotiates EEE to its link partner.
Step 4	no power efficient-ethernet auto Example: Device(config-if)# no power efficient-ethernet auto	Disables EEE on the specified interface.
Step 5	end Example: Device(config-if)# end	Exits interface configuration mode, and returns to privileged EXEC mode.
Step 6	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Monitoring EEE

Table 10: Commands for Displaying EEE Settings

Command	Purpose
show eee capabilities interface <i>interface-id</i>	Displays EEE capabilities for the specified interface.
show eee status interface <i>interface-id</i>	Displays EEE status information for the specified interface.
show eee counters interface <i>interface-id</i>	Displays EEE counters for the specified interface.

Following are examples of the **show eee** commands:

```
Device# show eee capabilities interface gigabitEthernet2/0/1
```

```
Gi2/0/1
EEE(efficient-ethernet): yes (100-Tx and 1000T auto)
Link Partner : yes (100-Tx and 1000T auto)
```

```
ASIC/Interface : EEE Capable/EEE Enabled
```

```
Device# show eee status interface gigabitEthernet2/0/1
```

```
Gi2/0/1 is up
EEE(efficient-ethernet): Operational
Rx LPI Status : Low Power
Tx LPI Status : Low Power
Wake Error Count : 0

ASIC EEE STATUS
Rx LPI Status : Receiving LPI
Tx LPI Status : Transmitting LPI
Link Fault Status : Link Up
Sync Status : Code group synchronization with data stream intact
```

```
Device# show eee counters interface gigabitEthernet2/0/1
```

```
LP Active Tx Time (10us) : 66649648
LP Transitioning Tx : 462
LP Active Rx Time (10us) : 64911682
LP Transitioning Rx : 153
```

Configuration Examples for Configuring EEE

This example shows how to enable EEE for an interface:

```
Device# configure terminal
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# power efficient-ethernet auto
```

This example shows how to disable EEE for an interface:

```
Device# configure terminal
Device(config)# interface gigabitethernet 1/0/1
Device(config-if)# no power efficient-ethernet auto
```

Additional References for EEE

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the <i>Interface and Hardware Commands</i> section of the <i>Command Reference (Catalyst 9200 Series Switches)</i> .

Feature History for Configuring EEE

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Fuji 16.9.2	Energy Efficient Ethernet	Energy Efficient Ethernet (EEE) is an IEEE 802.3az standard that is designed to reduce power consumption in Ethernet networks during idle periods.
Cisco IOS XE Cupertino 17.9.1	Energy Efficient Ethernet	This feature was implemented on C9200CX-12P-2X2G, C9200CX-8P-2X2G, and C9200CX-12T-2X2G models of the Cisco Catalyst 9200CX Series Switches, which were introduced in this release.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to [Cisco Feature Navigator](#).



CHAPTER 10

Configuring Power over Ethernet

- [Information About Power over Ethernet, on page 75](#)
- [How to Configure PoE and UPOE, on page 79](#)
- [Monitoring Power Status, on page 82](#)
- [Additional References for Power over Ethernet, on page 83](#)
- [Feature History for Power over Ethernet, on page 83](#)

Information About Power over Ethernet

For information on Configuring Perpetual PoE and 2-event Classification, refer *Network Powered Lighting Configuration Guide, Cisco IOS XE Fuji 16.9.x (Catalyst 9200 Switches)*

The following sections provide information about Power over Ethernet (PoE), the supported protocols, and standards and power management.

PoE and PoE+ Ports

A PoE-capable switch port automatically supplies power to one of these connected devices if the switch senses that there is no power in the circuit:

- A Cisco prestandard powered device (such as a Cisco IP phone)
- An IEEE 802.3af-compliant powered device
- An IEEE 802.3at-compliant powered device

A powered device can receive redundant power when it is connected to a PoE switch port and to an AC power source. The device does not receive redundant power when it is only connected to the PoE port.

Supported Protocols and Standards

The device uses the following protocols and standards to support PoE:

- **Cisco Discovery Protocol (CDP) with power consumption:** The powered device notifies the device of the amount of power it is consuming. The device does not reply to the power-consumption messages. The device can only supply power to or remove power from the PoE port.
- High-power devices can operate in low-power mode on the device that do not support power-negotiation CDP.

Cisco intelligent power management is backward-compatible with CDP with power consumption; the device responds according to the CDP message that it receives. CDP is not supported on third party-powered devices. Therefore, the device uses the IEEE classification to determine the power usage of the device.

- **IEEE 802.3af:** The major features of this standard are powered-device discovery, power administration, disconnect detection, and optional powered-device power classification.
- **IEEE 802.3at:** The PoE+ standard increases the maximum power that can be drawn by a powered device from 15.4 W per port to 30 W per port.

Powered-Device Detection and Initial Power Allocation

The switch detects a Cisco prestandard or an IEEE-compliant powered device when the PoE-capable port is in the no-shutdown state, PoE is enabled (the default), and the connected device is not powered by an AC adaptor.

After device detection, the switch determines the device's power requirements based on its type:

- The initial power allocation is the maximum amount of power that a powered device requires. The switch initially allocates this amount of power when it detects and powers the powered device. Because the switch receives CDP messages from the powered device, and because the powered device negotiates power levels with the switch through CDP power-negotiation messages, the initial power allocation might be adjusted.
- The switch classifies the detected IEEE device within a power consumption class. Based on the available power in the power budget, the switch determines if a port can be powered. The following table lists these levels.

Table 11: IEEE Power Classifications

Class	Maximum Power Level Required from the Device
0 (class status unknown)	15.4 W
1	4 W
2	7 W
3	15.4 W

The switch monitors and tracks requests for power and grants power only when it is available. The switch tracks the power budget (the amount of power available on the device for PoE). The switch also performs power-accounting calculations when a port is granted or denied power to keep the power budget up to date.

After power is applied to the port, the switch uses CDP to determine the *CDP-specific* power consumption requirement of the connected Cisco powered devices, which is the amount of power to allocate based on the CDP messages. The switch adjusts the power budget accordingly. Note that CDP does not apply to third-party PoE devices. The switch processes a request, and either grants or denies power. If the request is granted, the switch updates the power budget. If the request is denied, the switch ensures that the power to the port is turned off, generates a syslog message, and updates the LEDs. Powered devices can also negotiate with the switch for more power.

With PoE+, powered devices use IEEE 802.3at and LLDP power with medium-dependent interface (MDI) type, length, and value descriptions (TLVs) and power-via-MDI TLVs, for negotiating power up to 30 W.

Cisco prestandard devices and Cisco IEEE powered devices can use CDP or the IEEE 802.3 at power-via-MDI power-negotiation mechanism to request power levels up to 30 W.



Note The CDP-specific power consumption requirement is referred to as the *actual* power consumption requirement in the Cisco Catalyst Switches software configuration guides and command references.

If the switch detects a fault caused by an undervoltage, overvoltage, overtemperature, oscillator fault, or short-circuit condition, it turns off power to the port, generates a syslog message, and updates the power budget and LEDs.

Power Management Modes

The device supports these PoE modes:

- **auto:** The auto mode is the default setting. The device automatically detects if the connected device requires power. If the device discovers a powered device connected to the port, and if the device has enough power, it grants power, updates the power budget, and turns on power to the port on a first-come, first-served basis, and updates the LEDs. For LED information, see the hardware installation guide.

If the device has enough power for all the powered devices, they all come up. If enough power is available for all the powered devices connected to the device, power is turned on to all the devices. If enough PoE is not available, or if a device is disconnected and reconnected while other devices are waiting for power, it cannot be determined which devices are granted or are denied power.

If granting power exceeds the system's power budget, the device denies power, ensures that power to the port is turned off, generates a syslog message, and updates the LEDs. After power is denied, the device periodically rechecks the power budget and continues to attempt to grant the request for power.

If a device that is being powered by the device is then connected to wall power, the device might continue to power the device. The device might continue to report that it is still powering the device irrespective of whether the device is being powered by the device or receiving power from an AC power source.

If a powered device is removed, the device automatically detects the disconnect and removes power from the port. You can connect a nonpowered device without damaging it.

You can specify the maximum wattage that is allowed on the port. If the IEEE class maximum wattage of the powered device is greater than the configured maximum value, the device does not provide power to the port. If the device powers a powered device, but the powered device later requests, through CDP or LLDP messages, more than the configured maximum value, the device removes power to the port. The power that was allocated to the powered device is reclaimed into the global power budget. If you do not specify a wattage, the device delivers the maximum value. Use the **auto** setting on any PoE port.

- **static:** The device preallocates power to the port (even when no powered device is connected) and guarantees that power will be available for the port. The device allocates the port-configured maximum wattage, and the amount is never adjusted through the IEEE class or by CDP messages from the powered device. Because power is preallocated, any powered device that uses less than or equal to the maximum wattage, is guaranteed to be powered when it is connected to the static port. The port no longer participates in the first-come, first-served model.

However, if the powered device's IEEE class is greater than the maximum wattage, the device does not supply power to it. If the device learns through CDP messages that the powered device is consuming more than the maximum wattage, the device shuts down the powered device.

If you do not specify a wattage, the device preallocates the maximum value. The device powers the port only if it discovers a powered device. Use the **static** setting on a high-priority interface.

- **never**: The device disables powered-device detection and never powers the PoE port even if an unpowered device is connected. Use this mode only when you want to make sure that power is never applied to a PoE-capable port, making the port a data-only port.

For most situations, the default configuration (**auto** mode) works well, providing plug-and-play operation. No further configuration is required. However, configure a PoE port for a higher priority, to make it data only, or to specify a maximum wattage to disallow high-power powered devices on a port.

Power Monitoring and Power Policing

When policing of the real-time power consumption is enabled, the device takes action when a powered device consumes more power than the maximum amount allocated, which is also referred to as the *cutoff-power value*.

When PoE is enabled, the device senses and monitors the real-time power consumption of the connected powered device. This is called *power monitoring* or *power sensing*. The device also polices the power usage with the *power policing* feature.

Power monitoring is backward-compatible with Cisco intelligent power management and CDP-based power consumption. It works with these features to ensure that the PoE port can supply power to a powered device.

The device senses the real-time power consumption of the connected device as follows:

1. The device monitors the real-time power consumption by individual ports.
2. The device records the power consumption, including peak power usage, and reports this information through the CISCO-POWER-ETHERNET-EXT-MIB.
3. If power policing is enabled, the device polices power usage by comparing the real-time power consumption with the maximum power allocated to the device. The maximum power consumption is also referred to as the *cutoff power* on a PoE port.

If the device uses more than the maximum power allocation on the port, the device can either turn off the power to the port, or can generate a syslog message and update the LEDs (the port LED is now blinking amber) while still providing power to the device based on the device configuration. By default, power-usage policing is disabled on all the PoE ports.

If error recovery from the PoE error-disabled state is enabled, the device automatically takes the PoE port out of the error-disabled state after the specified amount of time.

If error recovery is disabled, you can manually re-enable the PoE port by using the **shutdown** and **no shutdown** interface configuration commands.

4. If policing is disabled, no action occurs when the powered device consumes more than the maximum power allocation on the PoE port, which could adversely affect the device.

Power Consumption Values

You can configure the initial power allocation and the maximum power allocation on a port. However, these values are the configured values that determine when the device should turn on or turn off power on the PoE port. The maximum power allocation is not the same as the actual power consumption of the powered device. The actual cutoff power value that the device uses for power policing is not equal to the configured power value.

When power policing is enabled, the device polices the power usage *at the switch port*, where the power consumption is greater than that by the device. When you manually set the maximum power allocation, you must consider the power loss over the cable from the switch port to the powered device. The cutoff power is the sum of the rated power consumption of the powered device and the worst-case power loss over the cable.

We recommend that you enable power policing when PoE is enabled on your device. For example, for a Class 1 device, if policing is disabled and you set the cutoff-power value by using the **power inline auto max 6300** interface configuration command, the configured maximum power allocation on the PoE port is 6.3 W (6300 mW). The device provides power to the connected devices on the port if the device needs up to 6.3 W. If the CDP power-negotiated value or the IEEE classification value exceeds the configured cutoff value, the device does not provide power to the connected device. After the device turns on the power on the PoE port, the device does not police the real-time power consumption of the device, and the device can consume more power than the maximum allocated amount, which could adversely affect the device and the devices connected to the other PoE ports.

How to Configure PoE and UPOE

The following tasks describe how you can configure PoE and UPOE.

Configuring a Power Management Mode on a PoE Port



Note When you make PoE configuration changes, the port that are being configured drops power. Depending on the new configuration, the state of the other PoE ports and the state of the power budget, the port might not be powered up again. For example, port 1 is in the auto and on state, and you configure it for static mode. The device removes power from port 1, detects the powered device, and repowers the port. If port 1 is in the auto and on state, and you configure it with a maximum wattage of 10 W, the device removes power from the port and then redetects the powered device. The device repowers the port only if the powered device is a class 1, class 2, or a Cisco-only powered device.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface interface-id Example: Device (config) # interface gigabitethernet2/0/1	Specifies the physical port to be configured, and enters interface configuration mode.

	Command or Action	Purpose
Step 4	<p>power inline {auto [max <i>max-wattage</i>] never static [max <i>max-wattage</i>] }</p> <p>Example:</p> <pre>Device(config-if) # power inline auto</pre>	<p>Configures the PoE mode on the port. The following are the keywords:</p> <ul style="list-style-type: none"> • auto: Enables detection of powered devices. If enough power is available, automatically allocates power to the PoE port after device detection. This is the default setting. • max <i>max-wattage</i>: Limits the power allowed on the port. If no value is specified, the maximum is allowed. • never: Disables device detection and power to the port. <p>Note If a port has a Cisco-powered device connected to it, do not use the power inline never command to configure the port. A false link-up can occur, placing the port in the error-disabled state.</p> <ul style="list-style-type: none"> • static: Enables detection of powered devices. Preallocate (reserve) power for a port before the device discovers the powered device. The device reserves power for this port even when no device is connected, and guarantees that power will be provided upon device detection. <p>The device allocates power to a port configured in static mode before it allocates power to a port configured in auto mode.</p>
Step 5	<p>end</p> <p>Example:</p> <pre>Device(config-if) # end</pre>	Returns to privileged EXEC mode.
Step 6	<p> module <i>switch-number</i></p> <p>Example:</p> <pre>Device# show power inline</pre>	<p>Displays the PoE status for a device or a device stack, for the specified interface, or for a specified stack member.</p> <p>The module <i>switch-number</i> keywords are supported only on stacking-capable devices.</p>
Step 7	<p>copy running-config startup-config</p> <p>Example:</p> <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configuring Power Policing

By default, the device monitors the real-time power consumption of connected powered devices. You can configure the device to police the power usage. By default, policing is disabled.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet2/0/1	Specifies the physical port to be configured, and enters interface configuration mode.
Step 4	power inline police [action {log errdisable}] Example: Device(config-if)# power inline police	Configures the device to take one of these actions if the real-time power consumption exceeds the maximum power allocation on the port: <ul style="list-style-type: none"> • power inline police: Shuts down the PoE port, turns off power to it, and puts it in the error-disabled state. <p>Note You can enable error detection for the PoE error-disabled cause by using the errdisable detect cause inline-power global configuration command. You can also enable the timer to recover from the PoE error-disabled state by using the errdisable recovery cause inline-power interval <i>interval</i> global configuration command.</p> <ul style="list-style-type: none"> • power inline police action errdisable: Turns off power to the port if the real-time power consumption exceeds the maximum power allocation on the port. • power inline police action log: Generates a syslog message while still providing power to the port.

	Command or Action	Purpose
		If you do not enter the action log keywords, the default action shuts down the port and puts the port in the error-disabled state.
Step 5	exit Example: Device(config-if)# exit	Exits interface configuration mode, and returns to global configuration mode.
Step 6	Use one of the following: <ul style="list-style-type: none"> • errdisable detect cause inline-power • errdisable recovery cause inline-power • errdisable recovery interval <i>interval</i> Example: Device(config)# errdisable detect cause inline-power Device(config)# errdisable recovery cause inline-power Device(config)# errdisable recovery interval 100	(Optional) Enables error recovery from the PoE error-disabled state, and configures the PoE recovery mechanism variables. By default, the recovery interval is 300 seconds. interval <i>interval</i> : Specifies the time in seconds, to recover from the error-disabled state. The range is 30 to 86400.
Step 7	exit Example: Device(config)# exit	Returns to privileged EXEC mode.
Step 8	Use one of the following: <ul style="list-style-type: none"> • show power inline police • show errdisable recovery Example: Device# show power inline police Device# show errdisable recovery	Displays the power-monitoring status, and verifies the error recovery settings.
Step 9	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Monitoring Power Status

Use the following **show** commands to monitor and verify the PoE configuration.

Table 12: show Commands for Power Status

Command	Purpose
show power inline police	Displays power-policing data.

Additional References for Power over Ethernet

Related Documents

Related Topic	Document Title
For complete syntax and usage information pertaining to the commands used in this chapter.	See the "Interface and Hardware Commands" section in the <i>Command Reference Guide</i> .
For complete information on IEEE 802.3bt standard	See Cisco UPOE+: The Catalyst for Expanded IT-OT Convergence

Feature History for Power over Ethernet

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Fuji 16.9.2	Power over Ethernet (PoE)	<p>Power over Ethernet (PoE) allows the LAN switching infrastructure to provide power to an endpoint, called a powered device, over a copper Ethernet cable. The following types of end points can be powered through PoE:</p> <ul style="list-style-type: none"> • A Cisco prestandard powered device • An IEEE 802.3af-compliant powered device • An IEEE 802.3at-compliant powered device

Release	Feature	Feature Information
Cisco IOS XE Cupertino 17.9.1	Power over Ethernet (PoE)	This feature was implemented on C9200CX-12P-2X2G, C9200CX-8P-2X2G, and C9200CX-12T-2X2G models of the Cisco Catalyst 9200CX Series Switches, which were introduced in this release.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to [Cisco Feature Navigator](#).



CHAPTER 11

Configuring Perpetual PoE and Fast POE

- [Restrictions for Perpetual and Fast PoE, on page 85](#)
- [Information About Perpetual PoE, on page 85](#)
- [Fast POE, on page 86](#)
- [Configuring Perpetual and Fast PoE, on page 86](#)
- [Example: Configuring Perpetual and Fast PoE, on page 87](#)
- [Feature History for Perpetual PoE and Fast PoE, on page 87](#)

Restrictions for Perpetual and Fast PoE

The following restrictions apply to perpetual and fast PoE:

- Configuration of Fast PoE or Perpetual PoE has to be done before physically connecting any endpoint. Alternatively do a manual shut/no-shut of the ports drawing power.
- Power to the ports will be interrupted in case of MCU firmware upgrade and ports will be back up immediately after the upgrade.
- The CREE light powered device (PD) may flap at regular intervals if not configured with IP assigned from the DHCP server.
- If the PD does not support LLDP user can configure with either static or 2-event to receive required power as per the PD specification.

Information About Perpetual PoE

Perpetual PoE provides uninterrupted power to connected powered device even when a power sourcing equipment (PSE) switch is starting after a reload from executing the Cisco IOS software **reload** command.



Caution

Power to the ports will be interrupted in case of M3 or PSE firmware upgrade, and power to the ports will be backed up after Cisco IOS software starts.

Fast POE

This feature switches on power without waiting for IOS to boot up. When **poe-ha** is enabled on a particular port, the switch on a recovery after power failure, provides power to the connected endpoint devices within short duration before even the IOS forwarding starts up.

Configuring Perpetual and Fast PoE

To configure perpetual and Fast PoE, perform the following steps.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device (config)# interface gigabitethernet 2/0/1	Specifies the physical port to be configured, and enters interface configuration mode.
Step 4	power inline port perpetual-poe-ha Example: Device (config-if)# power inline port perpetual-poe-ha	Configures perpetual PoE. When you configure perpetual PoE on a port connected to a powered device, the powered device remains powered on during reload.
Step 5	power inline port poe-ha Example: Device (config-if)# power inline port poe-ha	Configures Fast PoE. When you configure Fast PoE, if the switch is power cycled, PD device powers on within 50-60 seconds of plugging into a power source without waiting for IOS to boot up.
Step 6	end Example: Device (config-if)# end	Returns to privileged EXEC mode.

Example: Configuring Perpetual and Fast PoE

This example shows how you can configure perpetual PoE on a switch:

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet2/0/1
Device(config-if)# power inline port perpetual-poe-ha
Device(config-if)# end
```

This example shows how you can configure fast PoE on the switch:

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet2/0/1
Device(config-if)# power inline port poe-ha
Device(config-if)# end
```

Feature History for Perpetual PoE and Fast PoE

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Table 13: Feature History for Perpetual PoE and Fast PoE

Releases	Feature	Feature Information
Cisco IOS XE Fuji 16.9.2	Perpetual and Fast PoE	Perpetual PoE provides uninterrupted power to a connected powered device even when the PSE switch is booting. Fast PoE switches on power without waiting for IOS to boot up.
Cisco IOS XE Cupertino 17.9.1	Perpetual PoE	This feature was implemented on C9200CX-12P-2X2G, C9200CX-8P-2X2G, and C9200CX-12T-2X2G models of the Cisco Catalyst 9200CX Series Switches, which were introduced in this release.

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to [Cisco Feature Navigator](#).



CHAPTER 12

Configuring 2-event Classification

- [Restrictions for 2-Event Classification, on page 89](#)
- [Information about 2-event Classification, on page 89](#)
- [Configuring 2-Event Classification, on page 89](#)
- [Example: Configuring 2-Event Classification, on page 90](#)
- [Feature History for 2-Event Classification, on page 90](#)

Restrictions for 2-Event Classification

The following restrictions apply to 2-event classification:

- Configuration of 2-event classification has to be done before physically connecting any endpoint. Alternatively, do a manual shut and unshut of the ports drawing power.
- Power to the ports is interrupted in case of microcontroller unit firmware upgrade and ports will be back up immediately after the upgrade.

Information about 2-event Classification

When a class 4 device gets detected, IOS allocates 30W without any CDP or LLDP negotiation. This means that even before the link comes up the class 4 power device gets 30W.

Also, on the hardware level the PSE does a 2-event classification which allows a class 4 PD to detect PSE capability of providing 30W from hardware, register itself and it can move up to PoE+ level without waiting for any CDP/LLDP packet exchange.

Once 2-event is enabled on a port, you need to manually shut/un-shut the port or connect the PD again to start the IEEE detection again. Power budget allocation for a class-4 device will be 30W if 2-event classification is enabled on the port, else it will be 15.4W.

Configuring 2-Event Classification

To configure a device for a 2-event classification, perform the steps given below.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>interface-id</i> Example: Device(config)# interface gigabitethernet2/0/1	Specifies the physical port to be configured, and enters interface configuration mode.
Step 4	power inline port 2-event Example: Device(config-if)# power inline port 2-event	Configures 2-event classification on the device.
Step 5	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

Example: Configuring 2-Event Classification

This example shows how you can configure 2-event classification:

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet2/0/1
Device(config-if)# power inline port 2-event
Device(config-if)# end
```

Feature History for 2-Event Classification

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Fuji 16.9.2	2-Event Classification	When a class-4 device gets detected, Cisco IOS allocates 30W without any CDP or LLDP negotiation. This means that even before the link comes up, the class-4 power device gets 30W.
Cisco IOS XE Cupertino 17.9.1	2-Event Classification	This feature was implemented on C9200CX-12P-2X2G, C9200CX-8P-2X2G, and C9200CX-12T-2X2G models of the Cisco Catalyst 9200CX Series Switches, which were introduced in this release.

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to [Cisco Feature Navigator](#).



CHAPTER 13

Configuring Auto SmartPorts

- [Restrictions for Auto SmartPorts, on page 93](#)
- [Information about Auto SmartPorts, on page 93](#)
- [How to Configure Auto SmartPorts, on page 96](#)
- [Configuration Examples for Auto SmartPorts, on page 97](#)
- [Feature History for Auto SmartPorts, on page 98](#)

Restrictions for Auto SmartPorts

- Although Auto SmartPort detects the Cisco switch it does not invoke the event trigger automatically. The event trigger needs to be manually invoked to map the switch to macros.

The **no macro auto global processing** command disables the Auto Smartport only. To disable the device classifier, use the **no device classifier** command.
- In a scenario where the user is authenticating for clients using the ASP macro and the macro includes commands that may trigger a session teardown or an internal configuration change, we observe that after authentication, the MAC address gets stuck in the drop state. The following are recommended workarounds to avoid this situation:
 - If the macro contains authentication commands, such as **authentication event server dead action authorize vlan *vlan-id*** and **authentication event no-response action authorize vlan *vlan-id***, remove the commands from the macro and configure them directly on the interface.
 - If the macro contains the **switchport access vlan *vlan-id*** command, use the Dynamic VLAN from the AAA server instead of configuring the VLAN via the macro.

Information about Auto SmartPorts

Auto SmartPort macros dynamically configure ports based on the device type detected on the port. When the switch detects a new device on a port, it applies the appropriate Auto SmartPorts macro. When a link-down event occurs on the port, the switch removes the macro. For example, when you connect a Cisco IP phone to a port, Auto SmartPorts automatically applies the Cisco IP phone macro. The Cisco IP phone macro enables quality of service (QoS), security features, and a dedicated voice VLAN to ensure proper treatment of delay-sensitive voice traffic.

Auto SmartPorts uses event triggers to map devices to macros. The most common event triggers are based on Cisco Discovery Protocol (CDP) messages received from connected devices. The detection of a device (Cisco IP phone, Cisco wireless access point, or Cisco router) invokes an event trigger for that device.

Link Layer Discovery Protocol (LLDP) is used to detect devices that do not support CDP. Other mechanisms used as event triggers include the 802.1X authentication result and MAC-address learned.

System built-in event triggers exist for various devices based mostly on CDP and LLDP messages and some MAC address. These triggers are enabled as long as Auto SmartPort is enabled.

You can configure user-defined trigger groups for profiles and devices. The name of the trigger group is used to associate a user-defined macro.

Auto SmartPort Macros

The Auto SmartPort macros are groups of CLI commands. Detection of devices on a port triggers the application of the macro for the device. System built-in macros exist for various devices, and, by default, system built-in triggers are mapped to the corresponding built-in macros. You can change the mapping of built-in triggers or macros as needed.

A macro basically applies or removes a set of CLIs on an interface based on the link status. In a macro, the link status is checked. If the link is up, then a set of CLIs is applied; if the link is down, the set is removed (the no format of the CLIs are applied). The part of the macro that applies the set of CLIs is termed macro. The part that removes the CLIs (the no format of the CLIs) are termed antimacro.

When a device is connected to an Auto SmartPort, if it gets classified as a lighting end point, it invokes the event trigger `CISCO_LIGHT_EVENT`, and the macro `CISCO_LIGHT_AUTO_SMARTPORT` is executed.

Commands run by CISCO_LIGHT_AUTO_SMARTPORT

When the macro is executed, it runs a series of commands on the switch.

The commands that are executed by running the macro `CISCO_LIGHT_AUTO_SMARTPORT` are:

- `switchport mode access`
- `switchport port-security violation restrict`
- `switchport port-security mac-address sticky`
- `switchport port-security`
- `power inline port poe-ha`
- `storm-control broadcast level 50.00`
- `storm-control multicast level 50.00`
- `storm-control unicast level 50.00`
- `spanning-tree portfast`
- `spanning-tree bpduguard enable`

Enabling Auto SmartPort



Note Auto SmartPorts are disabled by default.

To disable Auto SmartPort macros on a specific port, use the **no macro auto global processing** interface command before enabling Auto SmartPort globally.

To enable Auto SmartPort globally, use the **macro auto global processing** global configuration command.

To enable an Auto SmartPort, perform this task:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	device classifier Example: Device(config)# device classifier	Enables the device classifier. Use no device classifier command to disable the device classifier.
Step 4	macro auto global processing Example: Device(config)# macro auto global processing	Enables Auto SmartPorts on the switch globally. Use no macro auto global processing command to disable Auto SmartPort globally.
Step 5	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 6	show running-config Example: Device# show running-config	Verifies your entries.
Step 7	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

How to Configure Auto SmartPorts

The following section provides information about how to configure auto smartports.



Note Follow these guidelines when you are configuring Auto SmartPort Macros, performing active standby sync and configuring reload from primary to standby:

- Make sure there is no extra space in the configuration.
- Do not add extra parenthesis and tab in the configuration.
- Ensure that you do not use enter keyword more than required while configuring.

Configuring Mapping Between Event Triggers and Built-in Macros

To map an event trigger to a built-in macro, perform this task:

Before you begin

You need to enable Auto SmartPort macros globally. You need to perform this task when a Cisco switch is connected to the Auto SmartPort.

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	macro auto execute <i>event trigger</i> builtin <i>builtin macro name</i> Example: Device(config)# macro auto execute CISCO_SWITCH_EVENT builtin CISCO_SWITCH_AUTO_SMARTPORT	Specifies a user-defined event trigger and a macro name. This action configures mapping from an event trigger to a built-in Auto Smartports macro.
Step 4	macro auto trigger <i>event trigger</i> Example: Device(config)# macro auto trigger CISCO_SWITCH_EVENT	Invokes the user-defined event trigger.

	Command or Action	Purpose
Step 5	device <i>device_ID</i> Example: Device(config)# device cisco WS-C3560CX-8PT-S	Matches the event trigger to the device identifier.
Step 6	end Example: Device(config)# end	Returns to privileged EXEC mode.
Step 7	show shell triggers Example: Device# show shell triggers	Displays the event triggers on the switch.
Step 8	show running-config Example: Device# show running-config	Verifies your entries.
Step 9	copy running-config startup-config Example: Device# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuration Examples for Auto SmartPorts

The following sections provide configuration examples for Auto SmartPorts.

Example: Enabling Auto SmartPorts

The following example shows how you can enable an Auto SmartPort.

```
Device> enable
Device# configure terminal
Device(config)# device classifier
Device(config)# macro auto global processing
Device(config)# end
```

Example: Configuring Mapping Between Event Triggers and Built-In Macros

The following example shows how you can configure mapping between event triggers and built-in macros:

```
Device> enable
Device# configure terminal
Device(config)# macro auto execute CISCO_SWITCH_EVENT builtin CISCO_SWITCH_AUTO_SMARTPORT
Device(config)# macro auto trigger CISCO_SWITCH_EVENT
Device(config)# device cisco WS-C3560CX-8PT-S
Device(config)# end
```

Feature History for Auto SmartPorts

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Table 14: Feature History for Auto SmartPorts

Release	Feature	Feature Information
Cisco IOS XE Fuji 16.9.2	Auto SmartPorts	Auto SmartPort macros dynamically configure ports based on the device type detected on the port. When the switch detects a new device on a port, it applies the appropriate Auto SmartPorts macro.
Cisco IOS XE Cupertino 17.9.1	Auto SmartPorts	This feature was implemented on C9200CX-12P-2X2G, C9200CX-8P-2X2G, and C9200CX-12T-2X2G models of the Cisco Catalyst 9200CX Series Switches, which were introduced in this release.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to [Cisco Feature Navigator](#).



CHAPTER 14

Configuring COAP Proxy Server

- [Restrictions for the COAP Proxy Server, on page 99](#)
- [Information About the COAP Proxy Server, on page 99](#)
- [How to Configure the COAP Proxy Server, on page 100](#)
- [Configuration Examples for the COAP Proxy Server, on page 103](#)
- [Monitoring COAP Proxy Server, on page 107](#)
- [Feature History for COAP, on page 108](#)

Restrictions for the COAP Proxy Server

The following restrictions apply to COAP proxy server:

- Switch cannot advertise itself as CoAP client using ipv6 broadcast (CSCuw26467).
- Support for Observe Not Implemented.
- Blockwise requests are not supported. We handle block-wise responses and can generate block-wise responses.
- DTLS Support is for the following modes only RawPublicKey and Certificate Based.
- Switch does not act as DTLS client. DTLS for endpoints only.
- Endpoints are expected to handle and respond with CBOR payloads.
- Client side requests are expected to be in JSON.
- Switch cannot advertise itself to other Resource Directories as IPv6, due to an IPv6 broadcast issue.

Information About the COAP Proxy Server

The COAP protocol is designed for use with constrained devices. COAP works in the same way on constrained devices as HTTP works on servers in accessing information.

The comparison of COAP and HTTP is shown below:

- In the case of a webserver: **HTTP** is the protocol; **TCP** is the transport; and **HTML** is the most common information format transported.

- In case of a constrained device: **COAP** is the protocol; **UDP** is the transport; and **JSON/link-format/CBOR** is the popular information format.

COAP provides a means to access and control device using a similar **GET/POST** metaphor and restful API as in HTTP.

How to Configure the COAP Proxy Server

To configure the COAP proxy server, you can configure the COAP proxy and COAP endpoints in the configuration mode.

The commands are: **coap [proxy | endpoints]**.

Configuring the COAP Proxy

To start or stop the COAP proxy on the switch, perform the steps given below:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	coap proxy Example: Device(config)# coap proxy	Enters the COAP proxy sub mode. Note To stop the coap proxy and delete all configurations under coap proxy, use the no coap proxy command.
Step 4	security [none [[ipv4 ipv6] {ip-address ip-mask/prefix} list {ipv4-list name ipv6-list-name}] dtls [id-trustpoint {identity-trustpoint label}] [verification-trustpoint {verification-trustpoint} [ipv4 ipv6 {ip-address ip-mask/prefix} list {ipv4-list name ipv6-list-name}]]] Example: Device(config-coap-proxy)# security none ipv4 1.1.0.0 255.255.0.0	Takes the encryption type as argument. The two security modes supported are none and dtls <ul style="list-style-type: none"> • none: Indicates no security on that port. With security none, a maximum of 5 ipv4 and 5 ipv6 addresses can be associated. • dtls: The DTLS security takes RSA trustpoint and Verification trustpoint which are optional. Without Verification trustpoint it does the normal Public Key Exchange.

	Command or Action	Purpose
		<p>With security dtls, a maximum of 5 ipv4 and 5 ipv6 addresses can be associated.</p> <p>Note To delete all security configurations under coap proxy, use the no security command.</p>
Step 5	<p>max-endpoints {<i>number</i>}</p> <p>Example:</p> <pre>Device (config-coap-proxy) # max-endpoints 10</pre>	<p>(Optional) Specifies the maximum number of endpoints that can be learnt on the switch. The default value is 10. The range is 1 to 500.</p> <p>Note To delete all max-endpoints configured under coap proxy, use the no max-endpoints command.</p>
Step 6	<p>port-unsecure {<i>port-num</i>}</p> <p>Example:</p> <pre>Device (config-coap-proxy) # port-unsecure 5683</pre>	<p>(Optional) Configures a port other than the default 5683. The range is 1 to 65000.</p> <p>Note To delete all port configurations under coap proxy, use the no port-unsecure command.</p>
Step 7	<p>port-dtls {<i>port-num</i>}</p> <p>Example:</p> <pre>Device (config-coap-proxy) # port-dtls 5864</pre>	<p>(Optional) Configures a port other than the default 5684.</p> <p>Note To delete all dtls port configurations under coap proxy, use the no port-dtls command.</p>
Step 8	<p>resource-directory [ipv4 ipv6] {<i>ip-address</i>}]</p> <p>Example:</p> <pre>Device (config-coap-proxy) # resource-directory ipv4 192.168.1.1</pre>	<p>Configures a unicast upstream resource directory server to which the switch can act as a COAP client.</p> <p>With resource-directory, a maximum of 5 of ipv4 and 5 ipv6, ip addresses can be configured.</p> <p>Note To delete all resource directory configurations under coap proxy, use the no resource-directory command.</p>
Step 9	<p>list [ipv4 ipv6] {<i>list-name</i>}</p> <p>Example:</p> <pre>Device (config-coap-proxy) # list ipv4 trial_list</pre>	<p>(Optional) Restricts the IP address range where the lights and their resources can be learnt. Creates a named list of ip address/masks, to be used in the security [none dtls] command options above.</p> <p>With list, a maximum of 5 ip-lists can be configured, irrespective of ipv4 or ipv6. We</p>

	Command or Action	Purpose
		can configure a max of 5 ip addresses per ip-list. Note To delete any ip list on the COAP proxy server, use the no list [ipv4 ipv6] {list-name} command.
Step 10	start Example: Device (config-coap-proxy) # start	Starts the COAP proxy on this switch.
Step 11	stop Example: Device (config-coap-proxy) # stop	Stops the COAP proxy on this switch.
Step 12	exit Example: Device (config-coap-proxy) # exit	Exits the COAP proxy sub mode.
Step 13	end Example: Device (config) # end	Returns to privileged EXEC mode.

Configuring COAP Endpoints

To configure the COAP Proxy to support multiple IPv4/IPv6 static-endpoints, perform the steps given below:

Procedure

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	coap endpoint [ipv4 ipv6] {ip-address} Example: Device (config) # coap endpoint ipv4 1.1.1.1 Device (config) # coap endpoint ipv6 2001::1	Configures the static endpoints on the switch. <ul style="list-style-type: none"> • ipv4: Configures the IPv4 Static endpoints. • ipv6: Configures the IPv6 Static endpoints.

	Command or Action	Purpose
		Note To stop the coap proxy on any endpoint, use the no coap endpoint [ipv4 ipv6] {ip-address} command.
Step 4	exit Example: Device(config-coap-endpoint)# exit	Exits the COAP endpoint sub mode.
Step 5	end Example: Device(config)# end	Returns to privileged EXEC mode.

Configuration Examples for the COAP Proxy Server

Examples: Configuring the COAP Proxy Server

This example shows how you can configure the port number 5683 to support a maximum of 10 endpoints.

```
Device# coap proxy security none ipv4 2.2.2.2 255.255.255.0 port 5683 max-endpoints 10
```

This example shows how to configure COAP proxy on *ipv4 1.1.0.0 255.255.0.0* with **no** security settings.

```
Device(config-coap-proxy)# security ?
  dtls  dtls
  none  no security

Device(config-coap-proxy)# security none ?
  ipv4   IP address range on which to learn lights
  ipv6   IPv6 address range on which to learn lights
  list   IP address range on which to learn lights

Device(config-coap-proxy)# security none ipv4 ?
  A.B.C.D {/nn || A.B.C.D} IP address range on which to learn lights

Device(config-coap-proxy)# security none ipv4 1.1.0.0 255.255.0.0
```

This example shows how to configure COAP proxy on *ipv4 1.1.0.0 255.255.0.0* with **dtls id trustpoint** security settings.

```
Device(config-coap-proxy)# security dtls ?
  id-trustpoint DTLS RSA and X.509 Trustpoint Labels
  ipv4 IP address range on which to learn lights
  ipv6 IPv6 address range on which to learn lights
  list IP address range on which to learn lights

Device(config-coap-proxy)# security dtls id-trustpoint ?
  WORD Identity TrustPoint Label
```

```

Device(config-coap-proxy)# security dtls id-trustpoint RSA-TRUSTPOINT ?
verification-trustpoint Certificate Verification Label
<cr>

Device(config-coap-proxy)# security dtls id-trustpoint RSA-TRUSTPOINT

Device(config-coap-proxy)# security dtls ?
id-trustpoint DTLS RSA and X.509 Trustpoint Labels
ipv4 IP address range on which to learn lights
ipv6 IPv6 address range on which to learn lights
list IP address range on which to learn lights

Device(config-coap-proxy)# security dtls ipv4 1.1.0.0 255.255.0.0

```



Note For configuring **ipv4 / ipv6 / list**, the **id-trustpoint** and (optional) **verification-trustpoint**, should be pre-configured, else the system shows an error.

This example shows how to configure a Trustpoint. This is a pre-requisite for COAP **security dtls** with **id trustpoint** configurations.

```

ip domain-name myDomain
crypto key generate rsa general-keys exportable label MyLabel modulus 2048

Device(config)# crypto pki trustpoint MY_TRUSTPOINT
Device(ca-trustpoint)# rsakeypair MyLabel 2048
Device(ca-trustpoint)# enrollment selfsigned
Device(ca-trustpoint)# exit

Device(config)# crypto pki enroll MY_TRUSTPOINT
% Include the router serial number in the subject name? [yes/no]: no
% Include an IP address in the subject name? [no]: no
Generate Self Signed Router Certificate? [yes/no]: yes

```

This example shows how to configure COAP proxy on *ipv4 1.1.0.0 255.255.0.0* with **dtls verification trustpoint** (DTLS with certificates or verification trustpoints)

```

Device(config-coap-proxy)# security dtls ?
id-trustpoint DTLS RSA and X.509 Trustpoint Labels
ipv4 IP address range on which to learn lights
ipv6 IPv6 address range on which to learn lights
list IP address range on which to learn lights

Device(config-coap-proxy)# security dtls id-trustpoint ?
WORD Identity TrustPoint Label

Device(config-coap-proxy)# security dtls id-trustpoint RSA-TRUSTPOINT ?
verification-trustpoint Certificate Verification Label
<cr>

Device(config-coap-proxy)# security dtls id-trustpoint RSA-TRUSTPOINT verification-trustpoint
?

```

```
WORD Identity TrustPoint Label
```

```
Device(config-coap-proxy)# security dtls id-trustpoint RSA-TRUSTPOINT verification-trustpoint
CA-TRUSTPOINT ?
<Cr>
```

This example shows how to configure Verification Trustpoint. This is a pre-requisite for COAP **security dtls** with **verification trustpoint** configurations.

```
Device(config)# crypto pki import CA-TRUSTPOINT pkcs12 flash:hostA.p12 password cisco123
% Importing pkcs12...
Source filename [hostA.p12]?
Reading file from flash:hostA.p12
CRYPTO_PKI: Imported PKCS12 file successfully.
```

This example shows how to create a list named trial-list, to be used in the security [none | dtls] command options.

```
Device(config-coap-proxy)# list ipv4 trial_list
Device(config-coap-proxy-iplist)# 1.1.0.0 255.255.255.0
Device(config-coap-proxy-iplist)# 2.2.0.0 255.255.255.0
Device(config-coap-proxy-iplist)# 3.3.0.0 255.255.255.0
Device(config-coap-proxy-iplist)# exit
Device(config-coap-proxy)# security none list trial_list
```

This example shows all the negation commands available in the coap-proxy sub mode.

```
Device(config-coap-proxy)# no ?
ip-list          Configure IP-List
max-endpoints    maximum number of endpoints supported
port-unsecure    Specify a port number to use
port-dtls        Specify a dtls-port number to use
resource-discovery Resource Discovery Server
security         CoAP Security features
```

This example shows how you can configure multiple IPv4/IPv6 static-endpoints on the coap proxy.

```
Device(config)# coap endpoint ipv4 1.1.1.1
Device(config)# coap endpoint ipv4 2.1.1.1
Device(config)# coap endpoint ipv6 2001::1
```

This example shows how you can display the COAP protocol details.

```
Device# show coap version
```

```
CoAP version 1.0.0
RFC 7252
```

```
Device# show coap resources
```

```
Link format data =
</>
```

```

</1.1.1.6/cisco/context>
</1.1.1.6/cisco/actuator>
</1.1.1.6/cisco/sensor>
</1.1.1.6/cisco/lldp>
</1.1.1.5/cisco/context>
</1.1.1.5/cisco/actuator>
</1.1.1.5/cisco/sensor>
</1.1.1.5/cisco/lldp>
</cisco/flood>
</cisco/context>
</cisco/showtech>
</cisco/lldp>

```

Device# **show coap globals**

```

Coap System Timer Values :
  Discovery   : 120 sec
  Cache Exp  : 5 sec
  Keep Alive  : 120 sec
  Client DB   : 60 sec
  Query Queue: 500 ms
  Ack delay   : 500 ms
  Timeout     : 5 sec

```

```

Max Endpoints      : 10
Resource Disc Mode : POST

```

Device# **show coap stats**

```

Coap Stats :
Endpoints : 2
Requests  : 20
Ext Queries : 0

```

Device# **show coap endpoints**

List of all endpoints :

```

Code : D - Discovered , N - New
#   Status  Age(s)  LastWKC(s)  IP
-----
1   D       10       94          1.1.1.6
2   D        6       34          1.1.1.5

```

```

Endpoints - Total : 2 Discovered : 2 New : 0

```

Device# **show coap dtls-endpoints**

#	Index	State	String	State	Value	Port	IP
1	3	SSL	OK	3	48969	20.1.1.30	
2	2	SSL	OK	3	53430	20.1.1.31	
3	4	SSL	OK	3	54133	20.1.1.32	
4	7	SSL	OK	3	48236	20.1.1.33	

This example shows all options available to debug the COAP protocol.

```

Device# debug coap ?

all          Debug CoAP all
database     Debug CoAP Database
errors       Debug CoAP errors
events       Debug CoAP events
packet       Debug CoAP packet
trace        Debug CoAP Trace
warnings     Debug CoAP warnings

```

Monitoring COAP Proxy Server

To display the COAP protocol details, use the commands in the following table:

Table 15: Commands to Display to COAP specific data

show coap version	Shows the IOS COAP version and the RFC information.
show coap resources	Shows the resources of the switch and those learnt by it.
show coap endpoints	Shows the endpoints which are discovered and learnt.
show coap globals	Shows the timer values and end point values.
show coap stats	Shows the message counts for endpoints, requests and external queries.
show coap dtls-endpoints	Shows the dtls endpoint status.

Table 16: Commands to Clear COAP Commands

clear coap database	Clears the COAP learnt on the switch, and the internal database of endpoint information.
----------------------------	--

To debug the COAP protocol, use the commands in the following table:

Table 17: Commands to Debug COAP protocol

debug coap database	Debugs the COAP database output.
debug coap errors	Debugs the COAP errors output.
debug coap events	Debugs the COAP events output.
debug coap packets	Debugs the COAP packets output.
debug coap trace	Debugs the COAP traces output.
debug coap warnings	Debugs the COAP warnings output.
debug coap all	Debugs all the COAP output.



Note If you wish to disable the debugs, prepend the command with a **no** keyword.

Feature History for COAP

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Fuji 16.9.2	COAP	The COAP protocol is designed for use with constrained devices. COAP works in the same way on constrained devices as HTTP works on servers in accessing information.
Cisco IOS XE Cupertino 17.9.1	COAP	This feature was implemented on C9200CX-12P-2X2G, C9200CX-8P-2X2G, and C9200CX-12T-2X2G models of the Cisco Catalyst 9200CX Series Switches, which were introduced in this release.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to [Cisco Feature Navigator](#).



CHAPTER 15

Configuring an External USB Bluetooth Dongle

- [Restrictions for Configuring an External USB Bluetooth Dongle](#) , on page 109
- [Information About External USB Bluetooth Dongle](#), on page 109
- [How to Configure an External USB Bluetooth Dongle on a Switch](#), on page 110
- [Verifying Bluetooth Settings on a Device](#), on page 111
- [Feature History for Configuring an External Bluetooth Dongle](#), on page 111

Restrictions for Configuring an External USB Bluetooth Dongle

- Only Bluetooth version 4.0 is supported.
- External USB Bluetooth dongle is supported only on the Cisco Catalyst 9000 Series Switches that are configured within the IPv4 address range.
- In stacking mode, the external USB Bluetooth dongle needs to be enabled on an active switch.
- After a Stateful Switchover (SSO), the external USB Bluetooth dongle should be enabled on the new active switch interface.
- External USB Bluetooth dongle is not supported with the following configurations:
 - Quality of Service (QoS)
 - Access Control List (ACL)

Information About External USB Bluetooth Dongle

The connected external USB Bluetooth dongle acts as a Bluetooth host for external devices and serves as a management port on the switch. You can pair an external USB Bluetooth dongle with your Bluetooth-enabled external devices such as smart phone, laptop, or tablet.

External USB Bluetooth dongle is supported on switches that are configured both in standalone mode or in stacking mode.

Supported External USB Bluetooth Dongle

The following external USB Bluetooth dongles are supported:

- BTD-400 Bluetooth 4.0 Adapter by Kinivo
- Bluetooth 4.0 USB Adapter by Asus
- Mini Bluetooth Wireless USB 4.0 Dongle Adapter by Adnet
- Bluetooth 4.0 USB Adapter by Insignia

How to Configure an External USB Bluetooth Dongle on a Switch

To configure an external USB Bluetooth dongle on a switch, perform this procedure:

Procedure

Step 1 Connect an external USB Bluetooth dongle to the USB Type A port on the switch.

Note You can connect the external USB Bluetooth dongle either before powering up the device or when the device is running.

Step 2 On your switch, enter the global configuration mode and verify that the external USB Bluetooth dongle is connected to the switch:

```
Device> enable
Device# show platform hardware bluetooth
```

```
Controller:0:1a:7d:da:71:13
Type:Primary
Bus:USB
State:DOWN
Name:HCI Version:
```

Step 3 Enable Bluetooth interface using the **enable** command in interface configuration mode:

```
Device# configure terminal
Device(config)# interface bluetooth 0/4
Device(config-if)# enable
```

Step 4 Enter the **no shutdown** command to restart the Bluetooth interface automatically after a device reboot:

```
Device(config-if)# no shutdown
```

Step 5 Configure the pairing pin using the **bluetooth pin** *pin* command:

```
Device(config-if)# exit
Device(config)# bluetooth pin 1111
```

Step 6 Turn on the Bluetooth settings on your external device. On your external device, select the Bluetooth-enabled switch based on the hostname.

Step 7 Enable the network settings on your external device to allow it to connect to the internet.

Verifying Bluetooth Settings on a Device

This topic lists the commands used to monitor Bluetooth settings on a device.

Use the following commands in privileged EXEC mode to monitor Bluetooth settings.

Table 18: Commands to Monitor Bluetooth Settings on a Device

Command	Purpose
<code>show ip interface bluetooth 0/4</code>	Displays the usability status of a Bluetooth interface.
<code>show platform hardware bluetooth</code>	Displays information about a Bluetooth interface.
<code>show running include pin</code>	Displays the current Bluetooth pin.

Feature History for Configuring an External Bluetooth Dongle

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.12.1	Configuring an External Bluetooth Dongle	External USB Bluetooth dongle acts as a Bluetooth host for external devices and serves as a management port on the switch.
Cisco IOS XE Cupertino 17.9.1	Configuring an External Bluetooth Dongle	This feature was implemented on C9200CX-12P-2X2G, C9200CX-8P-2X2G, and C9200CX-12T-2X2G models of the Cisco Catalyst 9200CX Series Switches, which were introduced in this release.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to [Cisco Feature Navigator](#).



CHAPTER 16

Troubleshooting Interface and Hardware Components

- [Overview, on page 113](#)
- [Support Articles, on page 113](#)
- [Feedback Request, on page 114](#)
- [Disclaimer and Caution, on page 114](#)

Overview

This chapter provides links to documents authored by Cisco subject matter experts (SMEs). They aim to help you resolve technical issues without requiring a support ticket. If these documents are unable to resolve your issue, we recommend visiting the applicable [Cisco Community](#). There is a wealth of information and advice available from fellow Cisco customers who may have experienced this issue already and provided a solution. If you are not able to find a resolution on the Community, it may be best that you raise a support ticket at [Cisco Support](#). In cases where a support ticket has to be raised, these documents provide guidance about the data that should be collected and added to the support ticket. Specify the support document you referred, and TAC can create an improvement request with the document owner.

Support Articles

The documents in this section were created using specific software and hardware listed in the Components Used section of each article. However, this does not mean that they are limited to what is listed in Components Used, and generally remain relevant for later versions of software and hardware. Note that there could be some changes in the software or hardware that can cause commands to stop working, the syntax to change, or GUIs and CLIs to look different from one release to another.

The following are the support articles associated with this technology:

Document	Description
Unsupported Power Supplies on Catalyst 9200 and 9200L switches	This documents provides a list of incompatible power supplies for Catalyst 9200 and 9200L models of the Cisco Catalyst 9200 Series Switches.

Document	Description
Troubleshoot PoE on Catalyst 9000 Switches	This document describes the troubleshooting workflow for PoE (Power over Ethernet) on Catalyst 9000 PoE-capable switching platforms.
Troubleshoot MTU on Catalyst 9000 Series Switches	This document describes how to understand and troubleshoot MTU (Maximum Transmission Unit) on Catalyst 9000 series switches.
Understand why FCS Errors, Input Errors, or Packet Loss Can Occur when Connected to Multigigabit Ethernet Ports	This document describes why you encounter frame check sequence (FCS) errors, input errors, or packet loss with devices that connect to Multigigabit Ethernet (mGig) ports on Catalyst 9000 series switches due to interpacket gap (IPG) or interframe gap (IFG) tolerance.
Troubleshoot Port Flaps on Catalyst 9000 Series Switches	This document describes how to identify, collect useful logs, and troubleshoot problems that can occur with Port Flaps on Catalyst 9000 switches.

Feedback Request

Your input helps. A key aspect to improving these support documents is customer feedback. Note that these documents are owned and maintained by multiple teams within Cisco. If you find an issue specific to the document (unclear, confusing, information missing, etc):

- Provide feedback using the **Feedback** button located at the right panel of the corresponding article. The document owner will be notified, and will either update the article, or flag it for removal.
- Include information regarding the section, area, or issue you had with the document and what could be improved. Provide as much detail as possible.

Disclaimer and Caution

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.