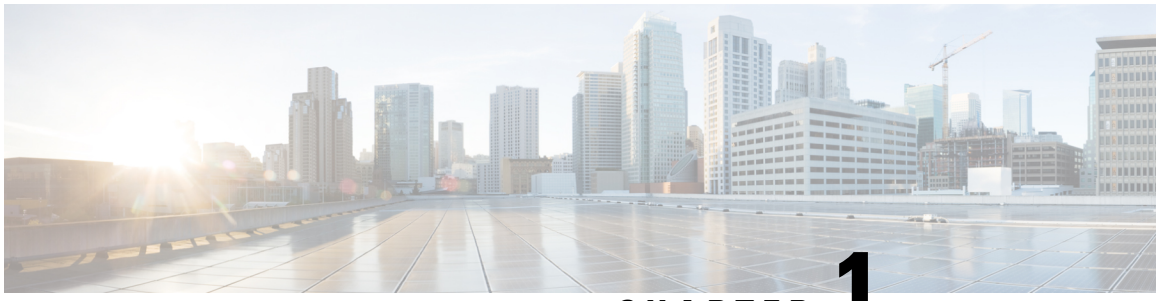
Interface and Hardware Components Configuration Guide, Cisco IOS XE Cupertino 17.7.x (Catalyst 9400 Switches)

First Published: 2021-12-07

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

CONTENTS



CHAPTER 1

Configuring Interface Characteristics

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.

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.

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.

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 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.

Uplink Ports

A supervisor module has 10 uplink ports, named 1 to 10. The first eight uplink ports, 1 to 8, use Small Form-Factor Pluggable (SFP) transceivers or SFP+ transceivers and uplink ports 9 and 10 use Quad Small Form-Factor Pluggable (QSFP) transceivers. Ports 1 to 8 are 10-Gigabit Ethernet ports that support both 10G and 1G transceivers. Ports 9 and 10 are the QSFP ports that support 40-Gigabit Ethernet uplinks. Additionally, Supervisor1XL25 supports 25-Gigabit Ethernet uplinks on ports 1 and 5. These ports 1 and 5 on Supervisor1XL25 use SFP28 transceivers to support 25-Gigabit mode.

By default, the 10-Gigabit Ethernet ports 1 to 8 are enabled.

Uplink Ports on Cisco Catalyst 9400 Series Supervisor1XL25 Module

The 10 uplink ports on Supervisor XL25 are grouped into two groups to support different speed configurations.

Port Group 1 supports 10G on ports 1 to 4; 25G on port 1 and 40G on port 9.

Port Group 2 supports 10G on ports 5 to 8; 25G on port 5 and 40G on port 10.

See the following table for port groupings and configurable speeds for Supervisor1XL25.

Table 1: Port Groupings on Cisco Catalyst 9400 Series Supervisor1XL25

Port Group	Port	Speed
Port Group 1 (ports 1,2,3,4,9)	1	10G or 25G
	2 - 4	10G
	9	40G
Port Group 2 (ports 5,6,7,8,10)	5	10G or 25G
	6 - 8	10G
	10	40G

Speeds 10G, 25G and 40G are mutually exclusive per port group. You can enable any one speed on a port group, at any given time.

For example, if you enable 25G on port 1, all the other speeds in Port Group 1 are disabled. If you configure 40G on port 10, 25G and 10G are disabled on the remaining ports in Port Group 2.

Note In a dual supervisor configuration (High Availability scenario), the ports in Port Group 2 are inactive. Only the ports in Port Group 1 are active.

Examples

The following examples are commands on Supervisor1XL25 Module fitted on a 10-slot chassis.

The following command enables 25G on port 1:

```
Switch(config)# interface TwentyFiveGigE5/0/1
Switch(config-if)# enable
Switch(config-if)#
```

The following command disables 25G on port 1:

```
Switch(config)# interface TwentyFiveGigE5/0/1
Switch(config-if)# no enable
*Jun  4 11:55:54.316: %TRANSCEIVER-6-REMOVED: R0/0: iomd: Transceiver module removed from
TwentyfiveGigabitEthernet5/0/1
```

The following command throws an error because it tries to configure 40G on port 9 when 25G is already configured on port 1:

```
Switch(config)# interface FortyGigabitE5/0/9
Switch(config-if)# enable
Twe5/0/1 currently configured with enable command - remove this before enabling on
FortyGigabitE5/0/9
```

Uplink Ports on Cisco Catalyst 9400 Series Supervisor2XL Module

The 8 uplink ports on Supervisor2XL are grouped into four groups to support different speed configurations.

Port Group 1 supports 25G on ports 1 to 4, and 100G on port 5, which is shared by ports 1 to 4.

Port Groups 2 to 4 supports 100G on ports 6 to 8.

In a redundant setup, ports 1 to 6 are used, and ports 7 and 8 are inactivate.

See the following table for port groupings and configurable speeds for Supervisor2XL.

Table 2: Port Groupings on Cisco Catalyst 9400 Series Supervisor2XL

Port Group	Port	Speed
Port Group 1 (ports 1,2,3,4,5)	1 - 4	25G
	5	100G
Port Group 2 (port 6)	6	100G
Port Group 3 (port 7)	7	100G
Port Group 4 (port 8)	8	100G

There are a total of 4 port groups operating at 100G speed, where ports 1 to 4 operating at 25G and port 5 operating at 100G are mutually exclusive.

Examples

The following examples are commands on Supervisor2 and Supervisor2XL module fitted on a 10-slot chassis.

The following command enables 100G on port 5 and disables port 1 to 4:

```
Switch(config)# interface HundredGigE5/0/5
Switch(config-if)# enable
Switch(config-if)#
May 25 11:18:01.586 PDT: %TRANSCEIVER-6-REMOVED: C5/0: iomd: Transceiver module removed
from TwentyFiveGigE5/0/1
May 25 11:18:01.590 PDT: %TRANSCEIVER-6-REMOVED: C5/0: iomd: Transceiver module removed
from TwentyFiveGigE5/0/2
May 25 11:18:01.593 PDT: %TRANSCEIVER-6-REMOVED: C5/0: iomd: Transceiver module removed
from TwentyFiveGigE5/0/3
May 25 11:18:01.596 PDT: %TRANSCEIVER-6-REMOVED: C5/0: iomd: Transceiver module removed
from TwentyFiveGigE5/0/4
May 25 11:18:05.767 PDT: %TRANSCEIVER-6-INSERTED: C5/0: iomd: transceiver module inserted
in HundredGigE5/0/5
```

The following command disables 100G on port 5 and enables port 1 to 4:

```
Switch(config)# interface HundredGigE5/0/5
Switch(config-if)# no enable
May 25 11:18:29.832 PDT: %TRANSCEIVER-6-REMOVED: C5/0: iomd: Transceiver module removed
from HundredGigE5/0/5
May 25 11:18:31.011 PDT: %TRANSCEIVER-6-INSERTED: C5/0: iomd: transceiver module inserted
in TwentyFiveGigE5/0/1
May 25 11:18:31.015 PDT: %TRANSCEIVER-6-INSERTED: C5/0: iomd: transceiver module inserted
in TwentyFiveGigE5/0/2
May 25 11:18:31.019 PDT: %TRANSCEIVER-6-INSERTED: C5/0: iomd: transceiver module inserted
```

```
in TwentyFiveGigE5/0/3
May 25 11:18:31.022 PDT: %TRANSCEIVER-6-INSERTED: C5/0: iomd: transceiver module inserted
in TwentyFiveGigE5/0/4
```

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.

Cisco Universal Power Over Ethernet (Cisco UPoE) extends the IEEE PoE+ standard to double the power per port to 60 watts.

Cisco Universal Power Over Ethernet Plus (Cisco UPoE+) combines the new IEEE 802.3bt standard and Cisco UPoE to increase the power per port to 90 watts. The 802.3bt-compliant Type 4 powered devices support up to 90watts.

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

Using the Switch USB Ports

The has two USB ports on the front panel — a USB mini-Type B console port and a USB Type A port.

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.

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, USB 2.0 and USB 3.0, 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). USB 3.0 is also called SuperSpeed USB, used for higher file transfer rates. 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.

USB 2.0 Host Port

The USB 2.0 host 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 device 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. To reenable 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.

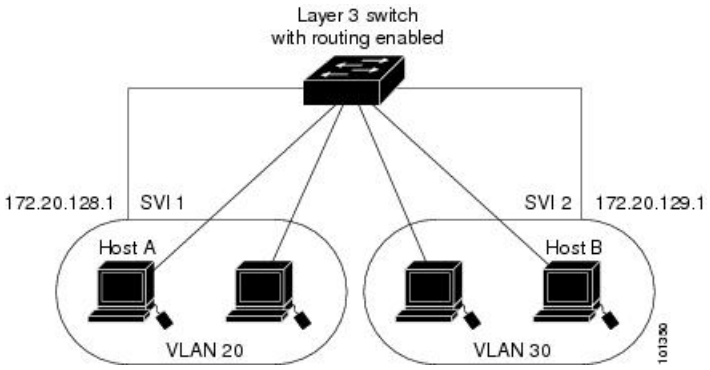
Note The **platform usb disable** command does not disable Bluetooth dongles connected to USB ports.

This command works on a device configured with Cisco StackWise Virtual.

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, module number, and device port number, and enter interface configuration mode.

- Type: Gigabit Ethernet (GigabitEthernet or gi) for 10/100/1000 Mbps Ethernet ports, 2.5-Gigabit Ethernet (TwoGigabitEthernet or tw) for 2.5 Gbps, 5-Gigabit Ethernet (FiveGigabitEthernet or fi) for 5 Gbps, 10-Gigabit Ethernet (TenGigabitEthernet or te) for 10 Gbps, 25-Gigabit Ethernet (TwentyFiveGigE or twe) for 25 Gbps, small form-factor pluggable (SFP) module Gigabit Ethernet and 10-Gigabit Ethernet interfaces and quad small-form-factor pluggable (QSFP) module 40-Gigabit Ethernet (FortyGigabitEthernet or fo) for 40 Gbps, and 100-Gigabit Ethernet (HundredGigE or hu) for 100 Gbps.
- Switch number: The number that identifies the given device. The number range is assigned the first time the device initializes.
- 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.

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 3: 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. (Not supported on the 25-Gigabit, 40-Gigabit, and 100-Gigabit interfaces.)
Duplex mode	Autonegotiate. (Not supported on the 25-Gigabit, 40-Gigabit, and 100-Gigabit interfaces.)
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.
Protected port	Disabled (Layer 2 interfaces only).
Port security	Disabled (Layer 2 interfaces only).

Feature	Default Setting
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

Ethernet interfaces on the switch operate at 10, 100, 1000 Mbps, 2.5 Gbps, 5 Gbps, 10 Gbps 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.

Speed and Duplex Configuration Guidelines

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

- Ethernet (10/100/1000-Mb/s) ports and multigigabit ethernet ports (2.5 Gb/s, 5Gb/s, 10 Gb/s) support all speed options and all duplex options (auto, half, and full). However, Gigabit Ethernet ports operating at 1000 Mb/s and above do not support half-duplex mode.
- 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 may or may not be up and this is expected.

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 **off**.

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. A routed port supports VLAN subinterfaces.

  VLAN subinterface: A 802.1Q VLAN subinterface is a virtual Cisco IOS interface that is associated with a VLAN id on a routed physical interface. The parent interface is a physical port. Subinterfaces can be created only on Layer 3 physical interfaces. A subinterface can be associated with different functionalities such as IP addressing, forwarding policies, Quality of Service (QoS) policies, and security policies. Subinterfaces divide the parent interface into two or more virtual interfaces on which you can

assign unique Layer 3 parameters such as IP addresses and dynamic routing protocols. The IP address for each subinterface should be in a different subnet from any other subinterface on the parent interface.

- 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**	Enables privileged EXEC mode.

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

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-id* **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** *string* **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** *interface-id* **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**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface range** {*port-range* | **macro** *macro_name*} **Example:** `Device(config)# interface range macro`	Specifies the range of interfaces (VLANs or physical ports) to be configured, and enter interface-range configuration mode. • 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 *port-range*, you must enter the interface type for each entry and enter spaces before and after the comma. • In a hyphen-separated *port-range*, you do not need to re-enter the interface type, but you must enter a space before the hyphen. **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.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**show interfaces** [*interface-id*] **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:** `Device# copy running-config startup-config`	(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:** `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	**define interface-range** *macro_name interface-range* **Example:**	Defines the interface-range macro, and saves it in NVRAM. • The *macro_name* is a 32-character maximum character string. • A macro can contain up to five comma-separated interface ranges. • Each *interface-range* must consist of the same port type. **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.

	Command or Action	Purpose
Step 4	**interface range macro** *macro_name* **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 *macro_name*. 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.

Note SFP+ modules operate only at 10G speed and full duplex. You can't configure the speed and duplex parameters for SFP+ modules.

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# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet1/0/3`	Specifies the physical interface to be configured, and enters interface configuration mode.
Step 4	**speed** {**10** | **100** | **1000** | **auto** [**10** | **100** | **1000** | **10000**] | **nonegotiate**} **Example:** `Device(config-if)# speed 10`	Enters the appropriate speed parameter for the interface: • Enter **10**, **100**, **1000**, or **10000** to set a specific speed for the interface. • Enter **auto** to enable the interface to autonegotiate speed with the connected device. If you specify a speed and also set the **auto** keyword, the port autonegotiates only at the specified speeds. • The **nonegotiate** keyword is available only for SFP module ports. SFP module ports operate only at 1000 Mb/s but can be configured to not negotiate if connected to a device that does not support autonegotiation.
Step 5	**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 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 7	**show interfaces** *interface-id* **Example:**	Displays the interface speed and duplex mode configuration.

	Command or Action	Purpose
	`Device# show interfaces gigabitethernet1/0/3`	
Step 8	**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:** `Device(config)# interface gigabitethernet1/0/1`	Specifies the physical interface to be configured, and enters interface configuration mode.
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:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config-if)# end`	
Step 6	**show flowcontrol interface** *interface-id* **Example:** `Device# show flowcontrol interface GigabitEthernet1/0/1`	Verifies the specified interface flow control settings.
Step 7	**show flowcontrol module** *slot* **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:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** {**gigabitethernet** *interface-id*} | {**vlan** *vlan-id*} | {**port-channel** *port-channel-number*} **Example:** `Device(config)# interface`	Specifies the interface to be configured as a Layer 3 interface, and enters interface configuration mode.

	Command or Action	Purpose
	`gigabitethernet1/0/2`	
Step 4	**no switchport** **Example:** `Device(config-if)# no switchport`	(For physical ports only) Enters Layer 3 mode.
Step 5	**ip address** *ip_address subnet_mask* **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** [*interface-id*]	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 1000 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_addresssubnet_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.

	Command or Action	Purpose
Step 7	**tunnel mode gre ip** **Example:** `Device(config)#tunnel mode gre ip`	Configures the tunnel mode.
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** *interface-id* **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.

	Command or Action	Purpose
Step 6	**show running config interface** *interface-id*	(Optional) Shows the running configuration. Verifies the configuration.
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.

	Command or Action	Purpose
Step 5	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Restarts an interface.
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 7	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.

Configuring USB Inactivity Timeout

The configurable inactivity timeout reactivates the RJ-45 console port if the USB console port is activated but no input activity occurs on it for a specified time period. When the USB console port is deactivated due to a timeout, you can restore its operation by disconnecting and reconnecting the USB cable.

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.

	Command or Action	Purpose
Step 4	**usb-inactivity-timeout switch** *switch_number timeout-minutes* **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, peform 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 reenable the USB ports.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits to privileged EXEC mode.
Step 5	**copy running-config startup-config** **Example:**	(Optional) Saves your entries in the configuration file.

	Command or Action	Purpose
	`Device# copy running-config startup-config`	

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 4: show Commands for Interfaces

Command	Purpose
show interfaces *interface-number* **downshift** [**module** *module-number*]	Displays the downshift status details of the specified interfaces and modules.
show interfaces *interface-id* **status** [**err-disabled**]	Displays interface status or a list of interfaces in the error-disabled state.
show interfaces [*interface-id*] **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.
show interfaces [*interface-id*] **description**	Displays the description configured on an interface or all interfaces and the interface status.
show ip interface [*interface-id*]	Displays the usability status of all interfaces configured for IP routing or the specified interface.
show interface [*interface-id*] **stats**	Displays the input and output packets by the switching path for the interface.
show interface [*interface-id*] **link**[**module** *number*]	Displays the up time and down time of an interface or all interfaces.
show interfaces *interface-id*	(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 [*interface-id*] [{**transceiver properties** | **detail**}] *module number*]	Displays physical and operational status about an SFP module.
show running-config interface [*interface-id*]	Displays the running configuration in RAM for the interface.

Command	Purpose
show version	Displays the hardware configuration, software version, the names and sources of configuration files, and the boot images.
show controllers ethernet-controller *interface-id* **phy**	Displays the operational state of the auto-MDIX feature on the interface.

Clearing and Resetting Interfaces and Counters

Table 5: clear Commands for Interfaces

Command	Purpose
clear counters [*interface-id*]	Clears interface counters.
clear interface *interface-id*	Resets the hardware logic on an interface.
clear line [*number* | **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 a Range of Interfaces

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 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 *Command Reference (Catalyst 9400 Series Switches)* .

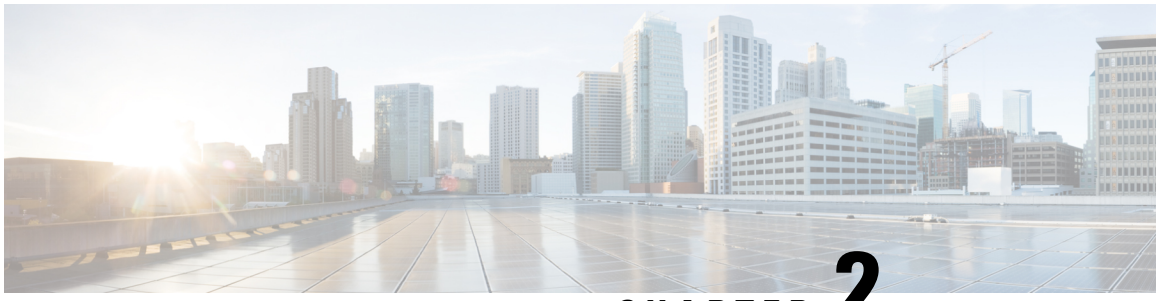
Feature History for Configuring Interface Characteristics

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 Everest 16.6.1	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 Everest 16.6.4	IEEE 802.3x Flow Control	The default value for **flowcontrol** interface configuration command was modified to **on** on all the models of the series.
Cisco IOS XE Amsterdam 17.2.1	Downlink support for Multi-Gigabit Ethernet Interfaces	Support to view downlink status of multi-gigabit ethernet interfaces was introduced.
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 17.7.1	Interface Characteristics	This feature was implemented on supervisor modules C9400X-SUP-2 and C9400X-SUP-2XL, 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 http://www.cisco.com/go/cfn.



CHAPTER 2

Configuring Auto-MDIX

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.

Information About Configuring 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 6: 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

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** *interface-id* **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 7: 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 Setting and Operational State on an Interface	Description
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.	*Command Reference (Catalyst 9400 Series Switches)*
For information about the power supplies.	*Cisco Catalyst 9400 Series Switches Hardware Installation Guide*

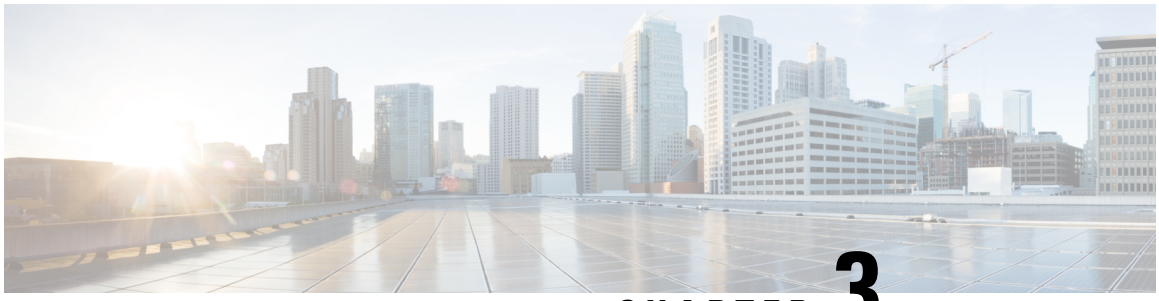
Feature History for Auto-MDIX

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 Everest 16.6.1	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 17.7.1	Auto-MDIX on an Interface	This feature was implemented on supervisor modules C9400X-SUP-2 and C9400X-SUP-2XL, 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 http://www.cisco.com/go/cfn.



CHAPTER 3

Configuring Ethernet Management Port

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.

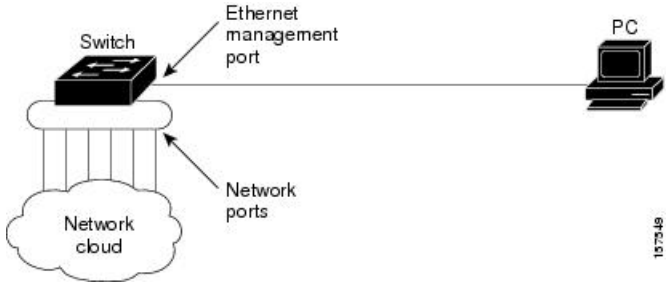
When managing a switch, connect the PC to the Ethernet Management port on Catalyst 9400 Series Switch.

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

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 with StackWise Virtual

Physically, the Ethernet management port needs to be connected from both active and standby switches to the uplink switch. Since the switches in a Cisco StackWise Virtual solution use a single management plane, the same IP address is applicable to both active and standby switches. After stateful switchover (SSO) between the active and standby switches, the Ethernet Management port on the active (previously standby) switch will link up and continue to support management functionalities.

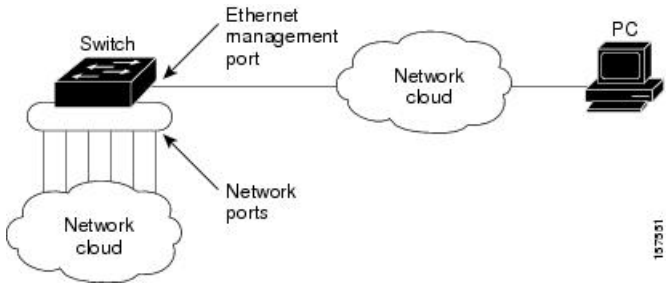
Note Any SSH, SCP, or Telnet sessions established by clients over the Ethernet management port IP address before stateful switchover to a new active switch in StackWise Virtual will be terminated and a new session has to be initiated after switchover.

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, 1000 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

Disabling and Enabling the Ethernet Management Port

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**interface gigabitethernet0/0** **Example:** `Device(config)# interface gigabitethernet0/0`	Specifies the Ethernet management port in the CLI.
Step 3	**shutdown** **Example:** `Device(config-if)# shutdown`	Disables the Ethernet management port.
Step 4	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Enables the Ethernet management port.
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 *System Management* section of this guide.
Bootloader commands	See the *System Management Commands* section of the *Command Reference (Catayst 9400 Series Switches)*

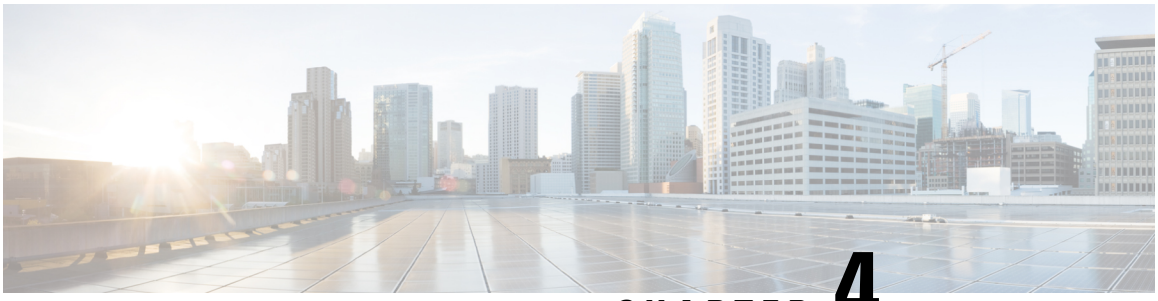
Feature History for Ethernet Management Port

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 Everest 16.6.1	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 17.7.1	Ethernet Management Port	This feature was implemented on supervisor modules C9400X-SUP-2 and C9400X-SUP-2XL, 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 http://www.cisco.com/go/cfn.



CHAPTER 4

Checking Port Status and Connectivity

Check Connected Modules

The Catalyst 9400 series switch is a modular system. You can see which modules are installed, and the MAC address ranges and version numbers for each module, by entering the show module command. Use the *mod_num* argument to specify a particular module number and display detailed information on that module.

This example shows how to check the status for all modules on your switch:

```
Device# show module

Chassis Type: C9410R

Mod Ports Card Type                                  Model          Serial No.
---+-----+--------------------------------------+--------------+--------------
1   48   48-Port UPOE w/ 24p mGig 24p RJ-45       C9400-LC-48UX    JAE2229053D
2   48   48-Port 5Gig/mGig 90W BT (RJ-45)         C9400-LC-48HN    JAE24530BF3
3   48   48-Port UPOE w/ 24p mGig 24p RJ-45       C9400-LC-48UX    JAE21280682
4   48   48-Port 5Gig/mGig 90W BT (RJ-45)         C9400-LC-48HN    JAE24241WAY
5   11   Supervisor 1 Module                      C9400-SUP-1      JAE22280PL8
6   11   Supervisor 1 Module                      C9400-SUP-1      JAE22280PHT
7   48   48-Port UPOE w/ 24p mGig 24p RJ-45       C9400-LC-48UX    JAE2229055N
8   48   48-Port UPOE w/ 24p mGig 24p RJ-45       C9400-LC-48UX    JAE22280DBU
9   48   48-Port UPOE w/ 24p mGig 24p RJ-45       C9400-LC-48UX    JAE22080BWS
10  48   48-Port UPOE w/ 24p mGig 24p RJ-45       C9400-LC-48UX    JAE230707YP
```

```
Mod MAC addresses                       Hw   Fw           Sw                  Status
---+--------------------------------+----+------------+-------------------+--------
1   BC26.C7A4.E738 to BC26.C7A4.E767 1.0  17.5.1r       17.05.01            ok
2   ECCE.13E2.B670 to ECCE.13E2.B69F 1.0  17.5.1r       17.05.01            ok
3   E4AA.5D59.A868 to E4AA.5D59.A897 1.0  17.5.1r       17.05.01            ok
4   A0B4.3982.43C0 to A0B4.3982.43EF 1.0  17.5.1r       17.05.01            ok
5   2C5A.0F1C.1EEC to 2C5A.0F1C.1EF6 2.0  17.5.1r       17.05.01            ok
6   2C5A.0F1C.1EF6 to 2C5A.0F1C.1F00 2.0  17.5.1r       17.05.01            ok
7   BC26.C7A4.D820 to BC26.C7A4.D84F 1.0  17.5.1r       17.05.01            ok
8   BC26.C772.E91C to BC26.C772.E94B 1.0  17.5.1r       17.05.01            ok
9   707D.B9C8.B5F8 to 707D.B9C8.B627 2.1  17.5.1r       17.05.01            ok
10  70EA.1ADB.7E74 to 70EA.1ADB.7EA3 3.0  17.5.1r       17.05.01            ok

Mod Redundancy Role     Operating Mode  Configured Mode  Redundancy Status
---+------------------+---------------+---------------+------------------
5   Active              sso             sso              Active
6   Standby             sso             sso              Standby Hot

Chassis MAC address range: 44 addresses from 2c5a.0f1c.1ec0 to 2c5a.0f1c.1eeb
```

Check Interface Status

You can view the summary or detailed information on the switch ports using the **show interface status** command. To see the summary information on all ports on the switch, enter the **show interface status** command with no arguments. Specify a particular module number to see information on the ports on that module only. Enter both the module number and the port number to see detailed information about the specified port.

To apply configuration commands to a particular port, you must specify the appropriate logical module.

This example shows how to display the status of all interfaces on a Catalyst 9400 series switch, including transceivers:

```
Switch# show interface status

Port      Name               Status       Vlan       Duplex  Speed Type
Gi1/0/1                      connected    1          a-full a-1000 10/100/1000BaseTX
Gi1/0/2                      connected    1          a-full a-1000 10/100/1000BaseTX
Gi1/0/3                      connected    1          a-full a-1000 10/100/1000BaseTX
Gi1/0/4                      notconnect   1            auto   auto 10/100/1000BaseTX
Gi1/0/5                      notconnect   1            auto   auto 10/100/1000BaseTX
Gi1/0/6                      notconnect   1            auto   auto 10/100/1000BaseTX
Gi1/0/7                      notconnect   1            auto   auto 10/100/1000BaseTX
Gi1/0/8                      notconnect   1            auto   auto 10/100/1000BaseTX
Gi1/0/9                      notconnect   1            auto   auto 10/100/1000BaseTX
Gi1/0/10                     notconnect   1            auto   auto 10/100/1000BaseTX
Gi1/0/11                     notconnect   1            auto   auto 10/100/1000BaseTX
Gi1/0/12                     notconnect   1            auto   auto 10/100/1000BaseTX
Gi1/0/13                     notconnect   1            auto   auto 10/100/1000BaseTX
Gi1/0/14                     notconnect   1            auto   auto 10/100/1000BaseTX
Gi1/0/15                     notconnect   1            auto   auto 10/100/1000BaseTX
Gi1/0/16                     notconnect   1            auto   auto 10/100/1000BaseTX
Gi1/0/17                     notconnect   1            auto   auto 10/100/1000BaseTX
Gi1/0/18                     notconnect   1            auto   auto 10/100/1000BaseTX
Gi1/0/19                     notconnect   1            auto   auto 10/100/1000BaseTX
Gi1/0/20                     notconnect   1            auto   auto 10/100/1000BaseTX
Gi1/0/21                     notconnect   1            auto   auto 10/100/1000BaseTX
Gi1/0/22                     notconnect   1            auto   auto 10/100/1000BaseTX
Gi1/0/23                     notconnect   1            auto   auto 10/100/1000BaseTX
Gi1/0/24                     notconnect   1            auto   auto 10/100/1000BaseTX
```

This example shows how to display the status of interfaces in error-disabled state:

```
Device# show interfaces status err-disabled
Port Name Status Reason
Fa9/4 err-disabled link-flap
informational error message when the timer expires on a cause
-----------------------------------------------------------------
5d04h:%PM-SP-4-ERR_RECOVER:Attempting to recover from link-flap err-disable state on Fa9/4
Switch#
```

Displaying PORT SET ENABLED LED Status

PORT SET ENABLED LED Status on Supervisor 1

There are four PORT SET ENABLED LEDs on the Supervisor 1 faceplate:

- One for port numbers 1 to 4, termed G1.
- One for port numbers 5 to 8, termed G2
- One for port number 9, termed G3
- One for port number 10, termed G4

Ports 1 to 8 are tengigabit ports and ports 9 and 10 are fortygigabit ports.

Standalone Supervisor 1

With a Standalone Supervisor, a single Supervisor is active and has ten ports. Group G1 and group G3 are mutually exclusive which means that either ports 1 to 4 are active or port 9 is active. Similarly, group G2 and group G4 are mutually exclusive; either ports 5 to 8 are active or port 10 is active. The status of the groups is decided by the configuration of the fortygigabit interfaces.

Displaying PORT SET ENABLED LED in a Standalone Supervisor 1 Mode

The following sample configuration enables the fortygigabit port number 10:

```
interface FortyGigabitEthernet4/0/9
end

interface FortyGigabitEthernet4/0/10
 enable
end
```

Following is a sample output of the **show hardware led** command:

```
SUPERVISOR: ACTIVE
PORT STATUS: (10) Te4/0/1:BLACK Te4/0/2:BLACK Te4/0/3:BLACK Te4/0/4:BLACK Te4/0/5:BLACK
Te4/0/6:BLACK Te4/0/7:BLACK Te4/0/8:BLACK Fo4/0/9:BLACK Fo4/0/10:BLACK

BEACON: BLACK

GROUP LED: UPLINK-G1:GREEN UPLINK-G2:BLACK UPLINK-G3:BLACK UPLINK-G4:GREEN
```

In this sample, you can see that group 4 is active (GREEN) and hence group 2 is inactive (BLACK). Since group 3 is not enabled and is inactive (BLACK), group 1 is active (GREEN)

High Availability or Dual Supervisor 1 Mode

In a dual supervisor mode, the Tengigabit ports numbered 1 to 4 (G1) and the Fortygigabit port numbered 9 (G3) are operational on both the supervisors. The other Tengigabit ports numbered 5 to 8 (G2) and the Fortygigabit port numbered 10 (G4) are disabled by default. Of the groups G1 and G3 which are mutually exclusive, either of the groups are active based on the configuration of the Fortygigabit port number 9.

Displaying PORT SET ENABLED LED in a Dual Supervisor 1 Mode

```
Switch#show run int fo4/0/9
Building configuration...

Current configuration : 52 bytes
!
interface FortyGigabitEthernet4/0/9
 enable
end

Switch#


SUPERVISOR: STANDBY
PORT STATUS: (10) Te3/0/1:BLACK Te3/0/2:BLACK Te3/0/3:BLACK Te3/0/4:BLACK Te3/0/5:BLACK
Te3/0/6:BLACK Te3/0/7:BLACK Te3/0/8:BLACK Fo3/0/9:BLACK Fo3/0/10:BLACK

BEACON: BLACK

GROUP LED: UPLINK-G1:GREEN UPLINK-G2:BLACK UPLINK-G3:BLACK UPLINK-G4:BLACK


SUPERVISOR: ACTIVE
PORT STATUS: (10) Te4/0/1:BLACK Te4/0/2:BLACK Te4/0/3:BLACK Te4/0/4:BLACK Te4/0/5:BLACK
Te4/0/6:BLACK Te4/0/7:BLACK Te4/0/8:BLACK Fo4/0/9:BLACK Fo4/0/10:BLACK

BEACON: BLACK

GROUP LED: UPLINK-G1:BLACK UPLINK-G2:BLACK UPLINK-G3:GREEN UPLINK-G4:BLACK
```

PORT SET ENABLED LED Status on Supervisor 2

There are five PORT SET ENABLED LEDs on the Supervisor 2 faceplate:

- One for port numbers 1 to 4, termed G1.
- One for port number 5, termed G2
- One for port number 6, termed G3
- One for port number 7, termed G4
- One for port number 8, termed G5

Standalone Supervisor 2

With a Standalone Supervisor, a single Supervisor is active and has eight ports. Group G1 and group G2 are mutually exclusive which means that either ports 1 to 4 are active or port 5 is active. In a redundant setup, groups G1, G2, and G3 are used, and groups G4 and G5 are inactivate.

Displaying PORT SET ENABLED LED in a Standalone Supervisor 2 Mode

The following sample configuration enables the hundredgigabit port number 5:

```
interface HundredGigE3/0/5
end

interface HundredGigE3/0/5
 enable
end
```

Following is a sample output of the **show hardware led** command:

```
SUPERVISOR: ACTIVE
PORT STATUS: (10) Twe3/0/1:BLACK Twe3/0/2:ACT_GREEN Twe3/0/3:BLACK Twe3/0/4:BLACK
Hu3/0/5:BLACK Hu3/0/6:ACT_GREEN Hu3/0/7:BLACK Hu3/0/8:BLACK
BEACON: BLACK
STATUS: GREEN
GROUP LED: UPLINK-G1:GREEN UPLINK-G2:BLACK UPLINK-G3:GREEN UPLINK-G4:GREEN UPLINK-G5:GREEN
```

In this sample, you can see that groups 1, 3, 4, and 5 are active (GREEN) and group 2 is inactive (BLACK).

High Availability or Dual Supervisor 2 Mode

In a dual supervisor mode, the TwentyFiveGigabit ports numbered 1 to 4 (G1) and the HundredGigabit port numbered 6 (G3) are operational on both the supervisors. The other HundredGigabit ports numbered 7 and 8 (G4 and G5) are disabled by default.

Displaying PORT SET ENABLED LED in a Dual Supervisor 2 Mode

```
Switch#show run interface HundredGigE3/0/5
Building configuration...

Current configuration : 43 bytes
!
interface HundredGigE3/0/5
 enable
end

Switch#


SUPERVISOR: ACTIVE
PORT STATUS: (10) Twe3/0/1:BLACK Twe3/0/2:ACT_GREEN Twe3/0/3:BLACK Twe3/0/4:BLACK
Hu3/0/5:BLACK Hu3/0/6:ACT_GREEN Hu3/0/7:BLACK Hu3/0/8:BLACK
BEACON: BLACK
STATUS: GREEN
GROUP LED: UPLINK-G1:GREEN UPLINK-G2:BLACK UPLINK-G3:GREEN UPLINK-G4:BLACK UPLINK-G5:BLACK

SUPERVISOR: STANDBY
PORT STATUS: (10) Twe4/0/1:ACT_GREEN Twe4/0/2:ACT_GREEN Twe4/0/3:BLACK Twe4/0/4:BLACK
Hu4/0/5:BLACK Hu4/0/6:ACT_GREEN Hu4/0/7:BLACK Hu4/0/8:BLACK
BEACON: BLACK
STATUS: GREEN
GROUP LED: UPLINK-G1:GREEN UPLINK-G2:BLACK UPLINK-G3:GREEN UPLINK-G4:BLACK UPLINK-G5:BLACK
```

Display MAC Addresses

In addition to displaying the MAC address range for a module using the **show module** command, you can display the MAC address table information of a specific MAC address or a specific interface in the switch using the **show mac address-table address** and **show mac address-table interface** commands.

This example shows how to display MAC address table information for all MAC addresses:

```
Switch# show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
All    0100.0ccc.cccc    STATIC      CPU
All    0100.0ccc.cccd    STATIC      CPU
All    0180.c200.0000    STATIC      CPU
All    0180.c200.0001    STATIC      CPU
All    0180.c200.0002    STATIC      CPU
All    0180.c200.0003    STATIC      CPU
All    0180.c200.0004    STATIC      CPU
All    0180.c200.0005    STATIC      CPU
All    0180.c200.0006    STATIC      CPU
All    0180.c200.0007    STATIC      CPU
All    0180.c200.0008    STATIC      CPU
All    0180.c200.0009    STATIC      CPU
All    0180.c200.000a    STATIC      CPU
All    0180.c200.000b    STATIC      CPU
All    0180.c200.000c    STATIC      CPU
All    0180.c200.000d    STATIC      CPU
All    0180.c200.000e    STATIC      CPU
All    0180.c200.000f    STATIC      CPU
All    0180.c200.0010    STATIC      CPU
All    0180.c200.0021    STATIC      CPU
All    ffff.ffff.ffff    STATIC      CPU
   1    188b.45eb.cc01    DYNAMIC     Gi1/0/1
Total Mac Addresses for this criterion: 22
Switch#
```

This example shows how to display MAC address table information for a specific interface:

```
Switch# show mac address-table interface Gi1/0/1
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
   1    188b.45eb.cc01    DYNAMIC     Gi1/0/1
Total Mac Addresses for this criterion: 1
Switch#
```

Using Telnet

You can access the switch command-line interface (CLI) using Telnet. In addition, Telnet allows you to access other devices in the network. You can have up to eight simultaneous Telnet sessions.

Before you can open a Telnet session to the switch, you must first set the IP address (and in some cases the default gateway) for the switch. For information about setting the IP address and default gateway, see the section on *Configuring the Swtich for the First Time.*

Note To establish a Telnet connection to a host by using the hostname, configure and enable DNS.

To establish a Telnet connection to another device on the network from the switch, enter this command:

```
Switch# telnet host [port]
```

This example shows how to establish a Telnet connection from the switch to the remote host named labsparc:

```
Switch# telnet labsparc
Trying 172.16.10.3...
Connected to labsparc.
Escape character is '^]'.
UNIX(r) System V Release 4.0 (labsparc)
login:
```

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.

With TDR, you can check the status of copper cables on the 48-port 10/100/1000 BASE-T modules for the Catalyst 9400 Series Switch. TDR detects a cable fault by sending a signal through the cable and reading the signal that is reflected back. All or part of the signal can be reflected back due to defects in the cable.

Note Category 5 cable has four pairs. Each pair can assume one of the following states: open (not connected), broken, shorted, or terminated. The TDR test detects all four states and displays the first three as “Fault” conditions, and displays the fourth as “Terminated”. Although the CLI output is shown, the cable length is displayed only if the state is “Faulty.”

TDR feature is supported on the following modules:

- C9400-LC-48U
- C9400-LC-48T
- C9400-LC-48P

TDR detects a cable fault by sending a signal along its wires. Depending on the reflected signal, it can determine roughly where the cable fault has occured. The variations on how TDR signal is reflected back determine the results on TDR. On Catalyst 9400 Series Switch, only two types of cable fault types are detected - OPEN or SHORT. We do display Terminated status in case cable is properly terminated and this is done for illustrative purpose.

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. In those instances, the port on the device should be administratively down before the start of the TDR test.
- 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.
- 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.
- TDR results might differ between runs on different Catalyst 9400 modules because of the resolution difference of TDR implementations. When this occurs, you should refer to an offline cable diagnosis tool.

Change the Logout Timer

The logout timer automatically disconnects a user from the switch when the user is idle for longer than the specified time. To set the logout timer, enter this command:

```
Switch(config-line)# exec-timeout minutes seconds
```

This command changes the logout timer value (a timeout value of 0 prevents idle sessions from being disconnected automatically).

Use the **no** keyword to return to the default value.

To set the logout for 10 minutes and 10 seconds, enter the following command:

```
Switch(config)# line console 0
Switch(config-line)# exec-timeout 10 10
```

To set no logout timer for console session:

```
Switch(config)# line console 0
Switch(config-line)# exec-timeout 0 0
```

Monitor User Sessions

You can display the currently active user sessions on the switch using the **show users** command. The command output lists all active console port and Telnet sessions on the switch.

To display the active user sessions on the switch, enter this command:

```
Switch# show users [all]
```

To disconnect an active user session on the switch, enter the following command:

```
Switch# disconnect { console | ip_address }
```

Example

This example shows the output of the show users command when local authentication is enabled for console and Telnet sessions (the asterisk [*] indicates the current session)

```
Switch# show users
Line User Host(s) Idle Location
* 0 con 0 idle 00:00:00
Interface User Mode Idle Peer Address
```

```
Switch# show users all
  Line    User  Host(s)  Idle        Location
* 0 con 0       idle    00:00:00
  1 vty 0               00:00:00
  2 vty 1               00:00:00
  3 vty 2               00:00:00
  4 vty 3               00:00:00
  5 vty 4               00:00:00
  Interface User Mode   Idle        Peer Address
Switch#
```

This example shows how to disconnect an active console port session and an active Telnet session:

```
Switch> disconnect console
Console session disconnected.
Console> (enable) disconnect tim-nt.bigcorp.com
Telnet session from tim-nt.bigcorp.com disconnected. (1)
Switch# show users
Session User Location
-------- ---------------- -------------------------
telnet jake jake-mac.bigcorp.com
* telnet suzy suzy-pc.bigcorp.com
Switch#
```

Using Ping

These sections describe how to use IP ping:

How Ping Works

The ping command allows you to verify connectivity to remote hosts. If you attempt to ping a host in a different IP subnetwork, you must define a static route to the network or configure a router to route between those subnets.

The ping command is configurable from normal executive and privileged EXEC mode. A ping returns one of the following responses:

- Normal response—The normal response (hostname is alive) occurs in 1 to 10 seconds, depending on the network traffic.
- Destination does not respond—If the host does not respond, a No Answer message is returned.
- Unknown host—If the host does not exist, an Unknown Host message is returned.
- Destination unreachable—If the default gateway cannot reach the specified network, a Destination Unreachable message is returned.
- Network or host unreachable—If there is no entry in the route table for the host or network, a Network or Host Unreachable message is returned.

To stop a ping in progress, press Ctrl-C.

Run Ping Command

To ping another device on the network from the switch, enter this command in normal executive and privileged EXEC mode:

```
Switch# ping host
```

Checks connectivity to a remote host.

This example shows how to ping a remote host from normal executive mode:

```
Switch# ping labsparc
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Switch#
```

```
Switch# ping 72.16.10.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 72.16.10.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Switch#
```

This example shows how to use a ping command in privileged EXEC mode to specify the number of packets, the packet size, and the timeout period:

```
Switch# ping
Protocol [ip]: ip
Target IP address: 1.1.1.1
Repeat count [5]: 10
Datagram size [100]: 100
Timeout in seconds [2]: 10
Extended commands [n]: n
Sweep range of sizes [n]: n
Type escape sequence to abort.
Sending 10, 100-byte ICMP Echos to 1.1.1.1, timeout is 10 seconds:
!!!!!!!!!!
Success rate is 100 percent (10/10), round-trip min/avg/max = 1/1/1 ms
Switch#
```

Using IP Traceroute

How IP Traceroute Works

IP traceroute allows you to identify the path that packets take through the network on a hop-by-hop basis. The command output displays all network layer (Layer 3) devices, such as routers, that the traffic passes through on the way to the destination.

Layer 2 switches can participate as the source or destination of the trace command but does not appear as a hop in the trace command output.

The trace command uses the time to live (TTL) field in the IP header to cause routers and servers to generate specific return messages. Traceroute starts by sending a User Datagram Protocol (UDP) datagram to the destination host with the TTL field set to 1. If a router finds a TTL value of 1 or 0, it drops the datagram and sends back an Internet Control Message Protocol (ICMP) Time-Exceeded message to the sender. Traceroute determines the address of the first hop by examining the source address field of the ICMP Time-Exceeded message.

To identify the next hop, traceroute sends a UDP packet with a TTL value of 2. The first router decrements the TTL field by 1 and sends the datagram to the next router. The second router sees a TTL value of 1, discards the datagram, and returns the Time-Exceeded message to the source. This process continues until the TTL is incremented to a value large enough for the datagram to reach the destination host or until the maximum TTL is reached.

To determine when a datagram reaches its destination, traceroute sets the UDP destination port in the datagram to a large value that the destination host is unlikely to be using. When a host receives a datagram with an unrecognized port number, it sends an ICMP Port Unreachable error message to the source. The Port Unreachable error message indicates to traceroute that the destination has been reached.

Perform IP Traceroute

To trace the path that packets take through the network, enter this command in EXEC or privileged EXEC mode:

Procedure

	Command or Action	Purpose
Step 1	**traceroute** [*protocol*] [*destination*]	Runs IP traceroute to trace the path that packets take through the network.

Example

This example shows how to use the traceroute command to display the route that a packet takes through the network to reach its destination:

```
Switch# traceroute ip ABA.NYC.mil
Type escape sequence to abort.
Tracing the route to ABA.NYC.mil (26.0.0.73)
1 DEBRIS.CISCO.COM (192.180.1.6) 1000 msec 8 msec 4 msec
2 BARRNET-GW.CISCO.COM (192.180.16.2) 8 msec 8 msec 8 msec
```

```
3 EXTERNAL-A-GATEWAY.STANFORD.EDU (192.42.110.225) 8 msec 4 msec 4 msec
4 BB2.SU.BARRNET.NET (192.200.254.6) 8 msec 8 msec 8 msec
5 SU.ARC.BARRNET.NET (192.200.3.8) 12 msec 12 msec 8 msec
6 MOFFETT-FLD-MB.in.MIL (192.52.195.1) 216 msec 120 msec 132 msec
7 ABA.NYC.mil (26.0.0.73) 412 msec 628 msec 664 msec
Switch#
```

Layer 2 Traceroute

The Layer 2 traceroute feature allows the switch to identify the physical path that a packet takes from a source device to a destination device. Layer 2 traceroute supports only unicast source and destination MAC addresses. It determines the path by using the MAC address tables of the switches in the path. When the switch detects a device in the path that does not support Layer 2 traceroute, the switch continues to send Layer 2 trace queries and lets them time out.

If you want the switch to trace the path from a host on a source device to a host on a destination device, the switch can identify only the path from the source device to the destination device. It cannot identify the path that a packet takes from the source host to the source device or from the destination device to the destination host.

Layer 2 Traceroute Usage Guidelines

These are the Layer 2 traceroute usage guidelines:

- CDP must be enabled on all the devices in the network. For Layer 2 traceroute to function properly, do not disable CDP.

  If any devices in the physical path are transparent to CDP, the switch cannot identify the path through these devices.

- All switches in the physical path must have IP connectivity. When a switch is reachable from another switch, you can test connectivity by using the ping command in privileged EXEC mode.

- The maximum number of hops identified in the path is ten.

- You can enter the **traceroute mac** or the **traceroute mac ip command** in privileged EXEC mode on a switch that is not in the physical path from the source device to the destination device. All switches in the path must be reachable from this switch.

- The **traceroute mac** command output shows the Layer 2 path only when the specified source and destination MAC addresses belong to the same VLAN. If you specify source and destination MAC addresses that belong to different VLANs, the Layer 2 path is not identified, and an error message appears.

- If you specify a multicast source or destination MAC address, the path is not identified, and an error message appears.

- If the source or destination MAC address belongs to multiple VLANs, you must specify the VLAN to which both the source and destination MAC addresses belong. If the VLAN is not specified, the path is not identified, and an error message appears.

- The **traceroute mac ip** command output shows the Layer 2 path when the specified source and destination IP addresses belong to the same subnet. When you specify the IP addresses, the switch uses Address Resolution Protocol (ARP) to associate the IP address with the corresponding MAC address and the VLAN ID.

- If an ARP entry exists for the specified IP address, the switch uses the associated MAC address and identifies the physical path.
- If an ARP entry does not exist, the switch sends an ARP query and tries to resolve the IP address. If the IP address is not resolved, the path is not identified, and an error message appears.

- When multiple devices are attached to one port through hubs (for example, multiple CDP neighbors are detected on a port), the Layer 2 traceroute feature is not supported. When more than one CDP neighbor is detected on a port, the Layer 2 path is not identified, and an error message appears.
- This feature is not supported in Token Ring VLANs.

Perform Layer 2 Traceroute

To display the physical path that a packet takes from a source device to a destination device, enter either one of these commands:

```
Switch# traceroute mac source-mac-address destination-mac-address
```

OR

```
Switch# traceroute mac ip source-ip destination-ip
```

The following examples show how to use the **traceroute mac** and **traceroute mac ip** commands to display the physical path that a packet takes through the network to reach its destination:

```
Switch# traceroute mac cc16.7eaa.7203 188b.45eb.cc64
Source cc16.7eaa.7203 found on Switch
1 Switch (1.1.1.1) : Vl1 => Gi1/0/1
Destination 188b.45eb.cc64 found on Switch
Layer 2 trace completed.
Switch#
```

```
Switch# traceroute mac ip 1.1.1.1 1.1.1.2 detail
Translating IP to mac .....
1.1.1.1 => cc16.7eaa.7203
1.1.1.2 => 188b.45eb.cc64

Source cc16.7eaa.7203 found on Switch[C9410R] (1.1.1.1)
1 Switch / C9410R / 1.1.1.1 :Gi1/0/1 [auto, auto]
Destination 188b.45eb.cc64 found on Switch[C9410R] (1.1.1.1)
Layer 2 trace completed.
Switch#
```

Configure ICMP

Internet Control Message Protocol (ICMP) provides many services that control and manage IP connections. ICMP messages are sent by routers or access servers to hosts or other routers when a problem is discovered with the Internet header. For detailed information on ICMP, refer RFC 792.

Enable ICMP Protocol Unreachable Messages

If the Cisco IOS software receives a nonbroadcast packet that uses an unknown protocol, it sends an ICMP Protocol Unreachable message back to the source.

Similarly, if the software receives a packet that it is unable to deliver to the ultimate destination because it knows of no route to the destination address, it sends an ICMP Host Unreachable message to the source. This feature is enabled by default.

To enable the generation of ICMP Protocol Unreachable and Host Unreachable messages, enter the following command in interface configuration mode:

```
Switch (config-if)# [no] ip unreachables
```

Use the **no** keyword to disable the ICMP destination unreachable messages.

Note If you enter the **no ip unreachables** command, you will break the path MTU discovery functionality. Routers in the middle of the network might be forced to fragment packets.

To limit the rate that Internet Control Message Protocol (ICMP) destination unreachable messages are generated, enter the following command:

```
Switch (config)# [no] ip icmp rate-limit unreachable [df]  milliseconds
```

Use the **no** keyword to remove the rate limit and reduce the CPU usage.

Enable ICMP Mask Reply Messages

Occasionally, network devices must know the subnet mask for a particular subnetwork in the internetwork. To obtain this information, devices can send ICMP Mask Request messages. These messages are responded to by ICMP Mask Reply messages from devices that have the requested information. The Cisco IOS software can respond to ICMP Mask Request messages if the ICMP Mask. Reply function is enabled.

To have the Cisco IOS software respond to ICMP mask requests by sending ICMP Mask Reply messages, enter the following command:

```
Switch (config-if)# [no] ip mask-reply
```

Use the **no** keyword to disable this functionality.

Feature History for Checking Port Status and Connectivity

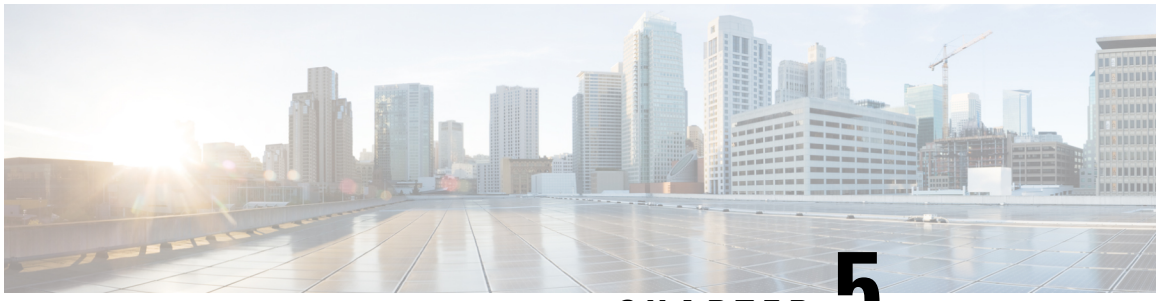
This table provides release and related information for features explained in this module.

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 Everest 16.6.1	Port Status and Connectivity Check	This feature includes the steps to check the status of modules,and interfaces; and also how to verify connectivity between devices within the network.

Release	Feature	Feature Information
Cisco IOS XE Fuji 16.8.1a	Command to display LED status	The **show hardware led** command was introduced to display the LED status.
Cisco IOS XE Cupertino 17.7.1	Port Status and Connectivity Check	This feature was implemented on supervisor modules C9400X-SUP-2 and C9400X-SUP-2XL, 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 http://www.cisco.com/go/cfn.



CHAPTER 5

Configuring LLDP, LLDP-MED, and Wired Location Service

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

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*] | **consumption <4000-60000> milliwatts**} interface configuration command. By default the PoE interface is in **auto** mode; If no value is specified, the maximum is allowed (60 W).

- 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 8: 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

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.

	Command or Action	Purpose
Step 3	**lldp run** **Example:** `Device(config)# lldp run`	Enables LLDP globally on the device.
Step 4	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet2/0/1`	Specifies the interface on which you are enabling LLDP, and enter 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`	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.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**lldp holdtime** *seconds* **Example:** `Device(config)# lldp holdtime 120`	(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)# lldp reinit 2`	(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)# lldp timer 30`	(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)# tlv-select`	(Optional) Specifies the LLDP TLVs to send or receive.
Step 7	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet2/0/1`	Specifies the interface on which you are enabling LLDP, and enter interface configuration mode.

	Command or Action	Purpose
Step 8	**lldp med-tlv-select** **Example:** `Device(config-if)# lldp med-tlv-select inventory management`	(Optional) Specifies the LLDP-MED TLVs to send or receive.
Step 9	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 10	**show lldp** **Example:** `Device# show lldp`	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 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 9: 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 enter 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`	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.

Configuring Network-Policy TLV

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**network-policy profile** *profile number* **Example:** `Device(config)# network-policy profile 1`	Specifies the network-policy profile number, and enter network-policy configuration mode. The range is 1 to 4294967295.
Step 4	{**voice** | **voice-signaling**} **vlan** [*vlan-id* {**cos** *cvalue* | **dscp** *dvalue*}] | [[**dot1p** {**cos** *cvalue* | **dscp** *dvalue*}] | **none** | **untagged**] **Example:** `Device(config-network-policy)# voice vlan 100 cos 4`	Configures the policy attributes: • **voice**—Specifies the voice application type. • **voice-signaling**—Specifies the voice-signaling application type. • **vlan**—Specifies the native VLAN for voice traffic. • *vlan-id*—(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). • **none**—(Optional) Do not instruct the IP telephone about the voice VLAN. The telephone uses the configuration from the telephone key pad.

	Command or Action	Purpose
		• **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** *interface-id* **Example:** `Device(config)# interface gigabitethernet2/0/1`	Specifies the interface on which you are configuring a network-policy profile, and enter interface configuration mode.
Step 7	**network-policy** *profile number* **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** *string* | **civic-location identifier** {*id* | **host**} | **elin-location** *string* **identifier** *id* | **custom-location identifier** {*id* | **host**} | **geo-location identifier** {*id* | **host**}} **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. • **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** *interface-id* **Example:**	Specifies the interface on which you are configuring the location information, and enter interface configuration mode.
Step 5	**location** {**additional-location-information** *word* | **civic-location-id** {*id* | **host**} |	Enters location information for an interface:

	Command or Action	Purpose
	elin-location-id *id* | **custom-location-id** {*id* | **host**} | **geo-location-id** {*id* | **host**} } **Example:** `Device(config-if)# location elin-location-id 1`	• **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. • **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`	Returns to privileged EXEC mode.
Step 7	Use one of the following: • **show location admin-tag** *string* • **show location civic-location identifier** *id* • **show location elin-location identifier** *id* **Example:** `Device# show location admin-tag` or `Device# show location civic-location identifier` or `Device# show location elin-location identifier`	Verifies the configuration.

	Command or Action	Purpose
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	**nmsp notification interval** {**attachment** | **location**} *interval-seconds* **Example:** `Device(config)# nmsp notification interval location 10`	Specifies the NMSP notification interval. **attachment**—Specifies the attachment notification interval. **location**—Specifies the location notification interval. *interval-seconds*—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`	Returns to privileged EXEC mode.

	Command or Action	Purpose
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

Configuring Network-Policy TLV: Examples

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

Commands for monitoring and maintaining LLDP, LLDP-MED, and wired location service.

Command	Description
clear lldp counters	Resets the traffic counters to zero.

Command	Description
clear lldp table	Deletes the LLDP neighbor information table.
clear nmsp 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 *entry-name*	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 [*interface-id*]	Displays information about interfaces with LLDP enabled. You can limit the display to a specific interface.
show lldp neighbors [*interface-id*] [**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.
show location admin-tag *string*	Displays the location information for the specified administrative tag or site.
show location civic-location identifier *id*	Displays the location information for a specific global civic location.
show location elin-location identifier *id*	Displays the location information for an emergency location
show network-policy profile	Displays the configured network-policy profiles.
show nmsp	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 *Interface and Hardware Commands* section in the *Command Reference (Catalyst 9400 Series Switches)*

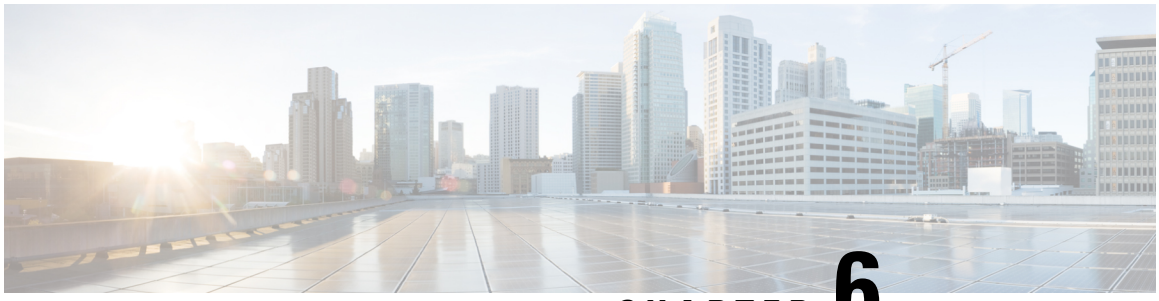
Feature History for LLDP, LLDP-MED, and Wired Location Service

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 Everest 16.6.1	Link Layer Discovery Protocol (LLDP), LLDP-MED, Wired Location Service	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-MED operates between endpoints and network devices. Wired Location Service lets you send tracking information of the connected devices to a Cisco Mobility Services Engine (MSE).
Cisco IOS XE Cupertino 17.7.1	Link Layer Discovery Protocol (LLDP), LLDP-MED, Wired Location Service	This feature was implemented on supervisor modules C9400X-SUP-2 and C9400X-SUP-2XL, 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 http://www.cisco.com/go/cfn.



CHAPTER 6

Configuring System MTU

Restrictions for System MTU

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

This table shows how the MTU values are applied.

Table 10: MTU Values

Configuration	system mtu command
Standalone switch	You can enter the **system mtu** command on a switch and it affects all ports on the switch. The range is from 1500 to 9216 bytes.

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

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`	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.
Step 2	**interface** *interface* **Example:** `Device(config)# interface gigabitethernet0/0`	Enters interface configuration mode.
Step 3	**ip mtu** *bytes* **Example:** `Device(config-if)# ip mtu 68`	Changes the IPv4 MTU size
Step 4	**ipv6 mtu** *bytes* **Example:** `Device(config-if)# ipv6 mtu 1280`	(Optional) Changes the IPv6 MTU size.
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`	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

```
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

```
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 *Interface and Hardware Commands* section in the *Command Reference (Catalyst 9400 Series Switches)*

Standards and RFCs

Standard/RFC	Title
RFC 8200	*Internet Protocol, Version 6 (IPv6) Specification*

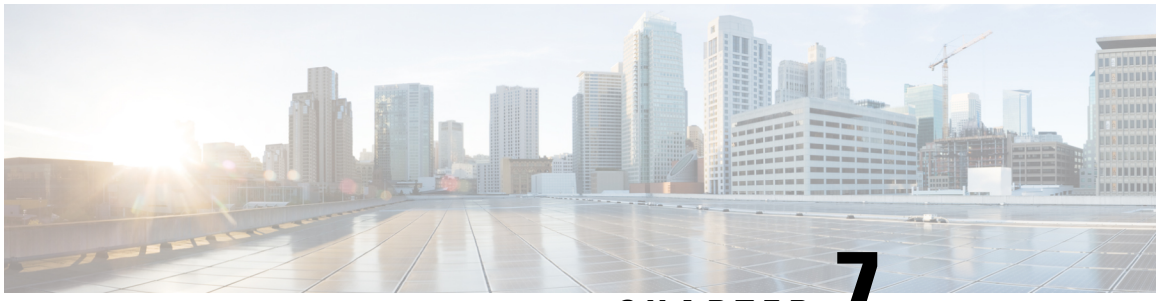
Feature History for System MTU

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 Everest 16.6.1	System MTU	System MTU defines the maximum transmission unit size for frames transmitted on all interfaces of a switch.
Cisco IOS XE Cupertino 17.7.1	System MTU	This feature was implemented on supervisor modules C9400X-SUP-2 and C9400X-SUP-2XL, 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 http://www.cisco.com/go/cfn.



CHAPTER 7

Configuring Per-Port MTU

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 sub interfaces and port-channel sub interfaces.

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.

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***typeswitch-number/slot-number/port-number* **Example:** `Device(config)# int FortyGigabitEthernet2/5/0/20`	Configures the interface and enters the 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`	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                        1500
Fo2/5/0/20                        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
```

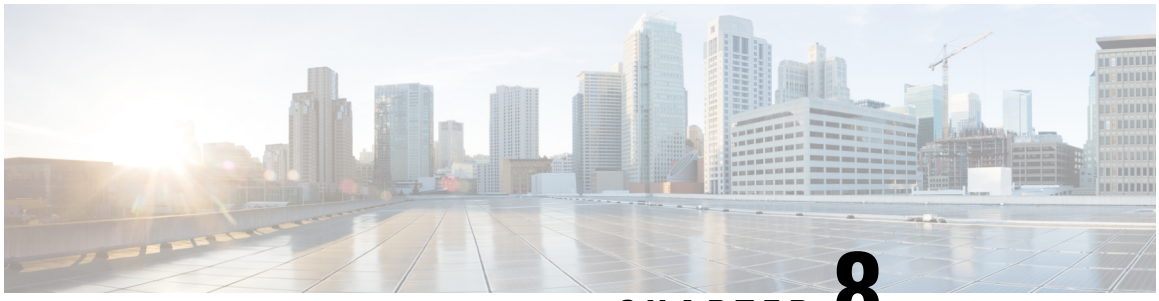
Feature History for Per-Port MTU

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 17.7.1	Per-Port MTU	This feature was implemented on supervisor modules C9400X-SUP-2 and C9400X-SUP-2XL, 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 http://www.cisco.com/go/cfn.



CHAPTER 8

Configuring EEE

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.
- You might want to enable the Link Layer Discovery Protocol (LLDP) for devices that require longer wakeup times before they are able to accept data on their receive paths. Doing so enables the device to negotiate for extended system wakeup times from the transmitting link partner.
- If a Multigigabit Ethernet port link is negotiated to 100 Mbps speeds, EEE will not initiate power-saving on the device.

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

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** *interface-id* **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Specifies the interface to be configured, and enter 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`	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# copy running-config startup-config`	

Monitoring EEE

Table 11: Commands for Displaying EEE Settings

Command	Purpose
show eee capabilities interface *interface-id*	Displays EEE capabilities for the specified interface.
show eee status interface *interface-id*	Displays EEE status information for the specified interface.
show eee counters interface *interface-id*	Displays EEE counters for the specified interface. **Note** Starting from Cisco IOS XE Gibraltar 16.12.1, **counters interface** *interface-id* command is no on line cards with Multigigabit (mGig) Etherne

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

```
Switch#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
```

```
Switch#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
```

```
Switch#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

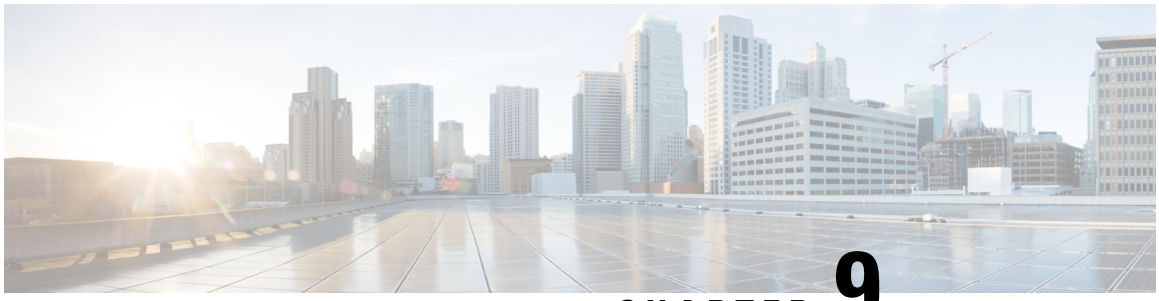
Feature History for Configuring EEE

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 Everest 16.6.1	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 Gibraltar 16.12.1	EEE on Multigigabit (mGig) Ethernet ports	Energy Efficient Ethernet was introduced on line cards with mGig Ethernet ports.
Cisco IOS XE Cupertino 17.7.1	Energy Efficient Ethernet	This feature was implemented on supervisor modules C9400X-SUP-2 and C9400X-SUP-2XL, 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 http://www.cisco.com/go/cfn.



CHAPTER 9

Configuring Power over Ethernet

Prerequisites for PoE Power Management

The following prerequisites apply to the PoE Power Management feature:

- The minimum supervisor field-programmable gate array (FPGA) version that is required for the feature to work is 19082605. Use the **showplatform** command to verify the supervisor FPGA version. If the FPGA version is earlier than 19082605, and the user tries to configure the **power inline auto-shutdown** command, the following message is displayed:

  ```
  This FPGA version does not support power inline auto shutdown feature.
  Please upgrade to FPGA from year 2019 and above.
  ```

  To upgrade the supervisor FPGA, use the **upgrade hw-programmable cpld filename bootflash: R0** command in EXEC mode.

- Interfaces can have the **power inline port priority** command configured, but for the PoE Power Management feature to work, the **power inline auto-shutdown** command must be configured in global configuration mode.

- Disable the following commands before performing ISSU:

  - **power inline auto-shutdown**
  - **power inline port priority** (on all the configured interfaces)

Information About Power over Ethernet

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
- An IEEE 802.3bt-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.
- **Cisco intelligent power management:** The powered device and the device negotiate through power-negotiation CDP messages for an agreed-upon power-consumption level. The negotiation allows a Cisco-powered device, which requires different power levels than its current allocation, to operate. The powered device first starts with its IEEE class power or 15.4 W (prestandard Cisco-powered device), and then negotiates power to operate at the appropriate power level. The device's consumption changes to the requested power mode only when it receives confirmation from the device.

  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.
- The **Cisco Universal Power Over Ethernet (UPOE)** feature provides the capability to source up to 60 W of power (2 x 30 W) over both signal and spare pairs of the RJ-45 Ethernet cable by using the Layer 2 power negotiation protocols such as CDP or LLDP. An LLDP and CDP request of 30 W and higher in the presence of the 4-wire Cisco proprietary spare-pair power type, length, and value descriptions (TLV) can provide power to the spare pair.

  When enabled in IEEE 802.3bt mode, Cisco UPOE devices function as 802.3bt Type 3 devices, supporting up to Class 6 (see Table 12: IEEE Power Classifications in the document) on every port.

Note The following UPOE linecards are IEEE 802.3bt-complaint PSE Type 3 devices:

- C9400-LC-48U
- C9400-LC-48UX

- **IEEE 802.3bt:** This new standard led to the introduction of new power sourcing equipment (PSE) that supports not just new capabilities but also compatibility with previous standards. Cisco introduced its 90-watt-capable line card on the Cisco Catalyst 9400 Series Switches that are in complete compliance with the IEEE 802.3bt standard and also support Cisco UPOE.

  This standard enables delivery of up to 90 W to a powered device over four pairs of Category 5e and above cables. It also introduces additional classes of PSEs and powered devices, class 5 to class 8, with PSE output power ranging between 45 W to 90 W, and the powered device input power ranging from 40 W to 71.3 W. This standard introduces new types of PSEs or powered devices, that is, Type 3(60 W) and Type 4 (90 W).

  The IEEE 802.3bt standard enables support for Dual Signature Powered Devices, Single Signature Powered Devices, and Single Pair Powered Devices. It also supports power demotion to handle scenarios where Type 4 Powered Device is connected to a Type 3 PSE.

  For more information, see the Additional References for Power over Ethernet section.

- **Cisco UPOE+**: Cisco UPOE+ combines the new IEEE 802.3bt standard and Cisco UPOE, which means Cisco UPOE+ switches are in complete compliance with the 802.3bt standard and also support all previous standards, such as 802.3af and IEE 802.3at, as well as Cisco UPOE. This feature provides the capability to source up to 90 W on the IEEE 802.3bt-compliant Type 4 devices.

  A Type 3 PSE can power up a Type 4 powered device through a power demotion to 60 W.

  Cisco IOS XE Release 16.12.1 introduces C9400-LC-48H, an 802.3bt-compliant Type 4 device.

  Cisco IOS XE Release 17.5.1 introduces C9400-LC-48HN, an 802.3bt-compliant Type 4 device.

  Some legacy Cisco powered devices (such as 7910, 7940, 7960 IP phones and AP350 wireless access points) are incompatible with Type 4 Power Supply Equipments (PSEs), as defined in the IEEE 802.3bt standard. If connected, the PSE will report a Tstart or Imax fault with each periodic attempt at providing power to the powered device. For continued use of these legacy Cisco powered devices, connect them to Cisco PoE+ or UPOE PSEs.

  Powered devices that do not meet the standard detection signature capacitance (such as CIVS-IPC-6000P) can be detected properly with PoE+ or Cisco UPOE devices running in UPOE mode, but may not be detected properly when running in 802.3bt mode.

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 12: 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
4	30 W
5	45 W
6	60 W
7	75 W
8	90 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, through CDP or LLDP. 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, and generates a syslog message. 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.

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.

  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 would exceed the system power budget, the device denies power, ensures that power to the port is turned off, and generates a syslog message. 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 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 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, the powered device can draw a maximum power based on what is allocated by the PSE. If the powered device consumes more than what is allocated, the port hits an Imax error and enters a fault condition.

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.

PoE Power Management

All ports are assigned a default PoE port priority based on the logical slot number of the linecard. Users can explicitly assign new priorities to the PoE ports by using the **power inline port priority** command in interface configuration mode. In a power shortage scenario, priority determines the order in which PoE ports will lose power. If the PoE Power Management feature is configured, PoE Ports with priority 7 (least priority) will shut down first and ports with priority 0 (highest priority) will shut down last, followed by line card shut down based on the autoLC shutdown priority. For more information, please refer the *Enabling Auto Line Card Shutdown* section of the *System Management Configuration Guide*

Ports in static mode have operational priority as 0, independent of the configured administration priority, so that during PoE load shedding, static ports shut down last. PoE ports are shut down before line cards are shut down.

The system can sustain an instantaneous drop of 9000 watts. We recommend that you do not assign more than 6000 watts to one PoE priority. If more than 6000 watts is configured for a PoE priotity level, a warning message is displayed, and if more than 9000 watts is configured for a PoE priotity level, a critical message is displayed.

The following table lists the slot numbers of the linecards along with the default PoE port priority:

Table 13: Default PoE Port Priority

Slot Number	Cisco Catalyst C9404R Switches	Cisco Catalyst C9407R Switches	Cisco Catalyst C9410R Switches
1	0	0	0
2	Supervisor	1	1
3	Supervisor	Supervisor	2
4	1	Supervisor	3
5	–	2	Supervisor
6	–	3	Supervisor
7	–	4	4
8	–	–	5
9	–	–	6
10	–	–	7

Cisco Universal Power Over Ethernet

Cisco Universal Power Over Ethernet (Cisco UPOE) is a Cisco-proprietary technology that extends the IEEE 802.3 at PoE standard to provide the capability to source up to 60 W of power over standard Ethernet cabling infrastructure (Class D or better) by using the spare pair of an RJ-45 cable (wires 4,5,7,8) with the signal pair (wires 1,2,3,6). Power on the spare pair is enabled when the switch port and end device mutually identify themselves as Cisco UPOE-capable using CDP or LLDP and the end device's requests for power to be enabled on the spare pair. When the spare pair is powered, the end device can negotiate up to 60 W of power from the switch using CDP or LLDP.

If the end device supports detection and classification on both signal and spare pairs, but does not support the CDP or LLDP extensions required for Cisco UPOE, a 4-pair forced mode configuration automatically enables power on both signal and spare pairs from the switch port.

You can upgrade the Cisco UPOE devices (C9400-LC-48U and C9400-LC-48UX) to support 802.3bt standard as a Type 3 power-sourcing device. A device can support Cisco UPOE and 802.3bt Type 3 on the same port. Note that while an 802.3bt-compliant Type 3 device and a Cisco UPOE device offer 60 W, they operate differently. 802.3bt-compliant devices mutually identify the maximum power requirements during physical classification (see Table 12: IEEE Power Classifications). An 802.3bt-compliant Type 3 powered device cannot ask for more power over LLDP than what is requested over the physical layer, which means that an 802.3bt-compliant Class 4 powered device cannot ask for more than 30 W using CDP or LLDP. However, an 802.3bt-compliant Class 6 powered device requests 60 W from the physical layer immediately before the data link layer is established.

In essence, 802.3at devices support ALT-A (signal pair) 30 W. Cisco UPOE devices support up to 60 W through CDP or LLDP negotiation. 802.3bt-compliant Type 3 4-pair devices can support up to 60 W across Alt-A and Alt-B (both pairs of wires) directly from the physical classification. 802.3bt-compliant Cisco UPOE PSEs support a powered device as physically requested. In addition, Cisco UPOE PSEs continue to support UPOE-powered devices. When you upgrade a Cisco UPOE device to 802.3bt mode, there is no change in the behavior of the UPOE-powered devices that are connected to the upgraded PSE.

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.
Step 4	**power inline** {**auto** [**max** *max-wattage*] | **never** | **static** [**max** *max-wattage*] | **consumption** *milli-watts-consumption* } **Example:** `Device(config-if)# power inline auto`	Configures the PoE mode on the port. The following are the keywords: • **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** *max-wattage*: Limits the power allowed on the port. The range for Cisco UPOE ports is 4000 to 60000 milliWatts (mW). If no value is specified, the maximum is allowed. • **never**: Disables device detection and power to the port. **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. • **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.

	Command or Action	Purpose
		• **consumption**: Sets the PoE consumption (in mW) of the powered device connected to a specific interface. The power consumption can range from 4000 to 90000 mW. To re-enable the automatic adjustment of consumption, either use the **no** keyword or specify 60000 mW. The device allocates power to a port configured in static mode before it allocates power to a port configured in auto mode.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show power inline** [[*interface-id*] [**detail**]] **Example:** `Device# show power inline`	Displays the PoE status for a device , for the specified interface.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Enabling Power on Signal and Spare Pairs

Note You do not have to perform this task if the line card on which the device is connected is in 802.3bt-compliance mode because the **power inline four-pair forced** command is redundant in the 802.3bt-compliance mode.

Do not peform this task if the end device cannot source inline power on the spare pair, or if the end device supports the CDP or LLDP extensions for Cisco UPOE.

Procedure

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

	Command or Action	Purpose
Step 2	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet2/0/1`	Specifies the physical port to be configured, and enters interface configuration mode.
Step 3	**power inline four-pair forced** **Example:** `Device(config-if)# power inline four-pair forced`	(Optional) Enables power on both signal and spare pairs from a switch port. **Note** This step is not required if the linecard on which the device is connected is in 802.3bt compliance mode.
Step 4	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

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** *interface-id* **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:

	Command or Action	Purpose
		• **power inline police**: Shuts down the PoE port, turns off power to it, and puts it in the error-disabled state. **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** *interval* global configuration command. • **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. 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`	Returns to global configuration mode.
Step 6	Use one of the following: • **errdisable detect cause inline-power** • **errdisable recovery cause inline-power** • **errdisable recovery interval** *interval* **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** *interval*: Specifies the time in seconds, to recover from the error-disabled state. The range is 30 to 86400.
Step 7	**exit** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# exit`	
Step 8	Use one of the following: • **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.

Configuring PoE Power Management

Before configuring the PoE port priority on an interface, the **power inline auto-shutdown** command must be enabled in global configuration mode. This command is disabled by default.

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	**power inline auto-shutdown** **Example:** `Device(config)# power inline auto-shutdown`	Enables auto shutdown control on PoE ports.
Step 4	**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 5	**power inline port priority** *value* **Example:** `Device(config-if)# power inline port priority 7`	Configures PoE port priority on the specified interface.
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Enable the 802.3bt Mode on Type 3 Cisco UPOE Modules

C9400-LC-48U and C9400-LC-48UX modules that support IEEE 802.3bt standard for Type 3 powered devices, are in 802.3at mode by default. You can enable 802.3bt mode on them using the **hw-module slot** *slot* **upoe-plus** command in global configuration mode. Note that the **hw-module slot** *slot* **upoe-plus** command power-cycles the module.

```
Device(config)# hw-module slot 4 upoe-plus
Performing oir to update poe fw on chassis 1 slot 4
Device#
*Mar 21 05:39:36.215: %IOSXE_OIR-6-REMSPA: SPA removed from subslot 4/0, interfaces disabled
```

Caution The **hw-module switch upoe-plus** command performs an online insertion and removal (OIR) on the module and the module will be out of service for the duration of the OIR.

You can revert to 802.3at mode using the **no** form of the **no hw-module slot** *slot* **upoe-plus** command.

Note C9400-LC-48H module is a Type 4 PSE that supports IEEE 802.3bt standard. C9400-LC-48H and C9400-LC-48HN is in 802.3bt mode by default. Therefore, the mode-conversion **hw-module slot** *slot* **upoe-plus** command is not supported on the C9400-LC-48H and C9400-LC-48HN modules.

Support for Noncompliant Powered Devices

You can allow a powered device, which is capable of drawing power on both pair sets, to draw more power than what is allowed on its physical layer, according to the IEEE Classification (Table 12: IEEE Power Classifications), using the **power inline auto** and **power inline static** commands.

The following example shows a Class 4 powered device configured to draw up to 40 W on the port it is connected to:

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet1/0/14
Device(config-if)# power inline static 40000
Device(config-if)# end

Device# show power inline upoe gigabitEthernet 1/0/14
Codes: DS - Dual Signature device, SS - Single Signature device
```

```
       SP - Single Pairset device

Interface   Admin  Type Oper-State      Power(Watts)    Class   Device Name
            State       Alt-A,B     Allocated Utilized  Alt-A,B
----------- ------ ---- ----------  --------- --------- ------- -----------

Gi1/0/14    static SS   on,on         40.0      36.7      4       Ieee PD
```

Monitoring Power Status

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

Table 14: show Commands for Power Status

Command	Purpose
show power inline police	Displays power-policing data.
show power inline [[*interface-id*] [**detail**]]	Displays PoE status for an interface on a switch.
show power inline consumption *interface-id*	Displays the PoE consumption for an interface.
show power inline upoe-plus [*interface-id*] [**module**]	Displays the PoE status for an interface that is enabled for 802.3bt-compliant mode.
show power inline priority *interface-id*	Displays the PoE states and priorities for an interface.
show power	Displays the available system power and the inline power of the interfaces. When power inline auto-shutdown is enabled, it also displays the total power allocation for each priority and shutdown threshold.

Examples

The following example displays the PoE status for an 802.3bt-enabled interface:

```
Device# show power inline upoe-plus gigabitEthernet 1/0/23

Codes: DS - Dual Signature device, SS - Single Signature device
       SP - Single Pairset device

Interface   Admin  Type Oper-State      Power(Watts)    Class   Device Name
            State       Alt-A,B     Allocated Utilized  Alt-A,B
----------- ------ ---- ----------  --------- --------- ------- -----------
Gi1/0/4     auto   SP   on            4.0       3.8       1       Ieee PD
Gi1/0/15    auto   SS   on,on         60.0      10.5      6       Ieee PD
Gi1/0/23    auto   DS   on,on         45.4      26.9      3,4     Ieee PD
```

The following table describes the significant fields shown in the display.

Table 15: show power inline upoe-plus Field Descriptions

Field	Description
Type	Type of Powered Device: Single Pairset device (SP), Single Signature device (SS), Dual Signature device (DS).
Oper-State	The state of each pair on the port.
Power Allocated	Power allocated to the port.
Power Utilized	Power consumed by the powered device on the port.
Class Alt-A, B	Signal and Spare pair respectively
Device Name	Name of the powered device as advertised by CDP.

The **show power inline detail** command is enhanced to display 802.3bt-complaint device information such as the Operational Status of the device, IEEE Class of the device, Physical Assigned Class, Allocated Power, (Power) Measured at the port.

Consider a scenario where a Class 5 Single Signature powered device sends a request through LLDP to lower the power allocated by PSE. Because of this, the power that is allocated drops to 30 W. The following is the output of the **show power inline detail** command in such a scenario:

```
Device# show power inline gigabitEthernet 1/0/29 detail

Interface: Gi1/0/29
Inline Power Mode: auto
Operational status (Alt-A,B): on,on
Device Detected: yes
Device Type: Ieee PD
Connection Check: SS
IEEE Class (Alt-A,B): 5
Physical Assigned Class (Alt-A,B): 5
Discovery mechanism used/configured: Ieee and Cisco
Police: off

Power Allocated
 Admin Value: 60.0
Power drawn from the source: 30.0
Power available to the device: 30.0
Allocated Power (Alt-A,B): 30.0

Actual consumption
Measured at the port(watts) (Alt-A,B): 10.5
Maximum Power drawn by the device since powered on: 10.5

Absent Counter: 0
Over Current Counter: 0
Short Current Counter: 0
Invalid Signature Counter: 0
Power Denied Counter: 0

Power Negotiation Used: IEEE 802.3bt LLDP
LLDP Power Negotiation       --Sent to PD--      --Rcvd from PD--
Power Type:                  Type 2 PSE           Type 2 PD
Power Source:                Primary              PSE
Power Priority:              low                  critical
PD 4PID:                     0                    1
```

```
Requested Power(W):          25.5                25.5
Allocated Power(W):          25.5                40.0
Requested Power ModeA(W):    0.0                 6.5
Allocated Power ModeA(W):    0.0                 25.5
Requested Power ModeB(W):    0.0                 13.0
Allocated Power ModeB(W):    0.0                 25.5
PSE Powering Status:         4 pair SS PD        Ignore
PD Powering Status:          Ignore              SS PD
PSE Power Pair ext:          Both Alternatives   Both Alternatives
DS Class Mode A ext:         SS PD               Class 2
DS Class Mode B ext:         SS PD               Class 4
SS Class ext:                Class 4             Class 5
PSE Type ext:                Type 3 PSE          Type 3 SS PD
PSE Max Avail Power:         51.0                51.2
PSE Auto Class Supp:         No                  No
PD Auto Class Req:           No                  No
PD Power Down Req:           No                  No
PD Power Down Time(sec):     0                   70

Four-Pair PoE Supported: Yes
Spare Pair Power Enabled: Yes
Four-Pair PD Architecture: Shared
```

The following example shows how a Dual-Signature powered device sends a request to lower the power allocated by the PSE:

```
Device# show power inline gigabitEthernet 1/0/23 detail

Interface: Gi1/0/23
Inline Power Mode: auto
Operational status (Alt-A,B): on,on
Device Detected: yes
Device Type: Ieee PD
Connection Check: DS
IEEE Class (Alt-A,B): 3,4
Physical Assigned Class (Alt-A,B): 3,4
Discovery mechanism used/configured: Ieee and Cisco
Police: off

Power Allocated
 Admin Value: 60.0
Power drawn from the source: 22.4
Power available to the device: 22.4
Allocated Power (Alt-A,B): 7.0,15.4
 Actual consumption
Measured at the port(watts) (Alt-A,B): 2.7,2.7
Maximum Power drawn by the device since powered on: 5.5
 Absent Counter: 0
Over Current Counter: 0
Short Current Counter: 0
Invalid Signature Counter: 0
Power Denied Counter: 0

Power Negotiation Used: IEEE 802.3bt LLDP
LLDP Power Negotiation       --Sent to PD--      --Rcvd from PD--
Power Type:                  Type 2 PSE           Type 2 PD
Power Source:                Primary              PSE
Power Priority:              low                  critical
PD 4PID:                     0                    1
Requested Power(W):          19.9                 0.0
Allocated Power(W):          19.9                 0.0
Requested Power ModeA(W):    6.5                  6.5
Allocated Power ModeA(W):    6.5                  13.0
Requested Power ModeB(W):    13.0                 13.0
Allocated Power ModeB(W):    13.0                 25.5
```

```
PSE Powering Status:         4 pair DS PD         Ignore
PD Powering Status:          Ignore               2 pair DS PD
PSE Power Pair ext:          Both Alternatives    Both Alternatives
DS Class Mode A ext:         Class 2              Class 3
DS Class Mode B ext:         Class 3              Class 4
SS Class ext:                DS PD                Class 5
PSE Type ext:                Type 3 PSE           Type 3 SS PD
PSE Max Avail Power:         51.0                 51.2
PSE Auto Class Supp:         No                   No
PD Auto Class Req:           No                   No
PD Power Down Req:           No                   No
PD Power Down Time(sec):     0                    70

Four-Pair PoE Supported: Yes
Spare Pair Power Enabled: Yes
Four-Pair PD Architecture: Independent
```

The following is a sample output of the **show power inline priority** command when the port priority is not configured on an interface. The admin priority, as shown below will be n/a and the operational priority will be the default priority based on the line card slot.

```
Device# show power inline priority gigabitEthernet 1/0/1

Interface  Admin  Oper       Admin      Oper
           State  State      Priority   Priority
     ---------- ------ ---------- ---------- --------
 Gi1/0/1    auto   on          n/a        0
```

The following is a sample output of the **show power inline priority** command when the port priority is configured as 5. With this configuration, both the admin and the operational priority will change to 5.

```
Device(config)# show power inline priority gigabitEthernet 1/0/1
Device(config-if)# power inline port priority 5
Device# show power inline priority gigabitEthernet 1/0/1

Interface   Admin   Oper       Admin      Oper
            State   State      Priority   Priority
  ---------- ------ ---------- ---------- --------
   Gi1/0/1    auto   on          5          5
```

The following is a sample output of the **show power inline priority** command when you configure a priority on static ports. The operation priority will change to 0, as static ports have the highest priority and will be the last to shutdown.

```
Device(config)# interface gigabitEthernet 1/0/1
Device(config-if)# power inline static
Device # show power inline priority gigabitEthernet 1/0/1

Interface  Admin  Oper       Admin      Oper
           State  State      Priority   Priority
---------- ------ ---------- ---------- --------
Gi1/0/1    static   on          5          0
```

The following table describes the significant fields shown in the display:

Table 16: show power inline priority Field Descriptions

Field	Description
Admin State	Administration mode: **auto**, **off**, **static**.
Oper State	Operating mode: **on**, **off**, **faulty**, **power-deny**.

Field	Description
Admin Priority	Administration priority level: 0 to 7.
Oper Priority	Operating priority level: 0 to 7.
Power Per Priority(Watts)	Power allocated to the PoE port priorities.

The following is the sample output of the **show power** command:

```
Device #show power
Power Fan States
Supply Model No Type Capacity Status 1 2
------ -------------------- ---- -------- ------------ -----------
PS1 C9400-PWR-3200AC ac 3200 W active good good
PS2 C9400-PWR-3200AC ac 3200 W active good good
PS3 C9400-PWR-3200AC ac 3200 W active good good
PS4 C9400-PWR-3200AC ac 3200 W active good good
PS5 C9400-PWR-3200AC ac 3200 W active good good
PS6 C9400-PWR-3200AC ac 3200 W active good good
PS7 C9400-PWR-3200AC ac 3200 W active good good
PS8 C9400-PWR-3200AC ac 3200 W active good good

PS Current Configuration Mode : Combined
PS Current Operating State : Combined

Power supplies currently active : 8
Power supplies currently available : 8

Power Summary Maximum
(in Watts) Used Available
------------ ------ ---------
System Power 2380 2380
Inline Power 4320 23220
------------ ------ ---------
Total 6700 25600

PoE POE Shutdown
Priority Allocation(Watts) Threshold(Watts)
---------- -------------------- ----------------------------
Priority-0 90 2475
Priority-1 0 2475
Priority-2 0 2475
Priority-3 0 2475
Priority-4 4230 6705
Priority-5 0 6705
Priority-6 0 6705
Priority-7 0 6705
```

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 *Command Reference Guide*.

Related Topic	Document Title
For complete information on IEEE 802.3bt standard	See Cisco UPOE+: The Catalyst for Expanded IT-OT Convergence

Feature History for Power over Ethernet

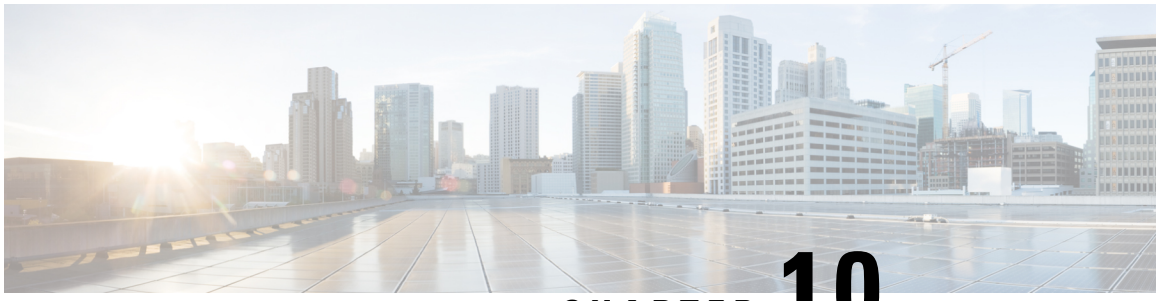
This table provides release and related information for features explained in this module.

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 Everest 16.6.1	Power over Ethernet (PoE)	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: • A Cisco prestandard powered device An IEEE 802.3af-compliant powered device An IEEE 802.3at-compliant powered device
Cisco IOS XE Gibraltar 16.11.1	Support for IEEE 802.3bt Type 3 PDs (up to 60 W)	The **hw-module slot upoe–plus** command was introduced to enable 802.3bt-compliant mode on the C9400-LC-48U, and C9400-LC-48UX line cards.
Cisco IOS XE Gibraltar 16.12.1	Support for IEEE 802.3bt Type 4 PDs (up to 90 W)	802.3bt-compliant Type 4 module, C9400-LC-48H was introduced.

Release	Feature	Feature Information
Cisco IOS XE Amsterdam 17.3.1	PoE Power Management	PoE Power Management allows port priority to be set on interfaces to determine which interface will shutdown first incase of a power outage.
Cisco IOS XE Bengaluru 17.5.1	Support for IEEE 802.3bt Type 4 PDs (up to 90 W)	802.3bt-compliant Type 4 module, C9400-LC-48HN was introduced.
Cisco IOS XE Cupertino 17.7.1	Power over Ethernet (PoE)	This feature was implemented on supervisor modules C9400X-SUP-2 and C9400X-SUP-2XL, 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 http://www.cisco.com/go/cfn.



CHAPTER 10

Configuring 2-event Classification

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/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.

Information about 2-event Classification

When a class 4 device gets detected, IOS allocates 15.4W without any CDP or LLDP negotiation.

Once 2-event config is enabled on a port, you need to manually shut or 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. This implies that even before the link comes up the class 4 PD will get 30W.

When 2-event is enabled on a port, at the hardware level the PSE does a 2-event classification which allows a class 4 PD to detect PSE capability of providing 30W through the hardware register. The PD can then move up to PoE+ level without waiting for any CDP or LLDP packet exchange.

Configuring 2-event Classification

To configure the switch 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** *interface-id* **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 switch.
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
```

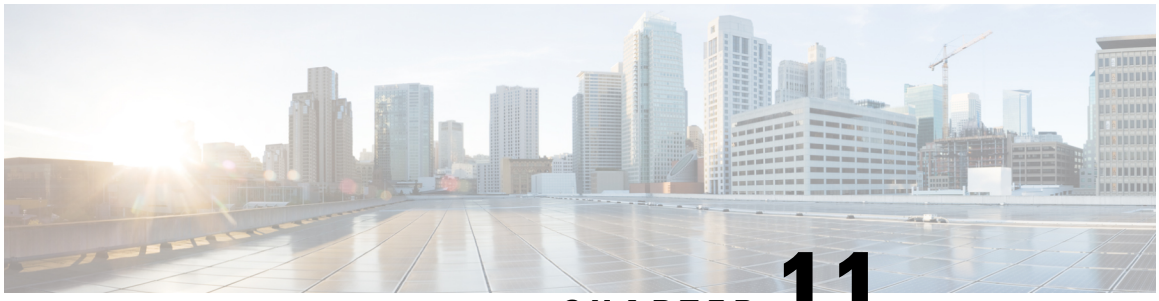
Feature History for 2-event Classification

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 Everest 16.6.1	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.
Cisco IOS XE Gibraltar 16.12.1	Change in allocated power	When a class 4 device gets detected, IOS allocates 15.4W without any CDP or LLDP negotiation.
Cisco IOS XE Cupertino 17.7.1	2-event classification	This feature was implemented on supervisor modules C9400X-SUP-2 and C9400X-SUP-2XL, 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 http://www.cisco.com/go/cfn.



CHAPTER 11

Configuring COAP Proxy Server

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:**	Takes the encryption type as argument. The two security modes supported are **none** and **dtls** • 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

	Command or Action	Purpose
	`Device(config-coap-proxy)# security none ipv4 1.1.0.0 255.255.0.0`	which are optional. Without Verification trustpoint it does the normal Public Key Exchange. With **security dtls**, a maximum of 5 ipv4 and 5 ipv6 addresses can be associated. **Note** To delete all security configurations under coap proxy, use the **no security** command.
Step 5	**max-endpoints** {*number*} **Example:** `Device(config-coap-proxy)#max-endpoints 10`	(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. **Note** To delete all max-endpoints configured under coap proxy, use the **no max-endpoints** command.
Step 6	**port-unsecure** {*port-num*} **Example:** `Device(config-coap-proxy)#port-unsecure 5683`	(Optional) Configures a port other than the default 5683. The range is 1 to 65000. **Note** To delete all port configurations under coap proxy, use the **no port-unsecure** command.
Step 7	**port-dtls** {*port-num*} **Example:** `Device(config-coap-proxy)#port-dtls 5864`	(Optional) Configures a port other than the default 5684. **Note** To delete all dtls port configurations under coap proxy, use the **no port-dtls** command.
Step 8	**resource-directory** [ipv4 | ipv6] {*ip-address*}] **Example:** `Device(config-coap-proxy)#resource-directory ipv4 192.168.1.1`	Configures a unicast upstream resource directory server to which the switch can act as a COAP client. With **resource-directory**, a maximum of 5 of ipv4 and 5 ipv6, ip addresses can be configured. **Note** To delete all resource directory configurations under coap proxy, use the **no resource-directory** command.
Step 9	**list** [**ipv4** | **ipv6**] {*list-name*} **Example:** `Device(config-coap-proxy)#list ipv4`	(Optional) Restricts the IP address range where the lights and their resources can be learnt. Creates a named list of ip address/masks, to

	Command or Action	Purpose
	`trial_list`	be used in the **security** [**none** | **dtls**] command options above. With **list**, a maximum of 5 ip-lists can be configured, irrespective of ipv4 or ipv6. We 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:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
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. • **ipv4** - Configures the IPv4 Static endpoints. • **ipv6** - Configures the IPv6 Static endpoints. **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.

```
#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     SSLOK       3           48969   20.1.1.30
2     2     SSLOK       3           53430   20.1.1.31
3     4     SSLOK       3           54133   20.1.1.32
4     7     SSLOK       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 17: 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 18: 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 19: 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.

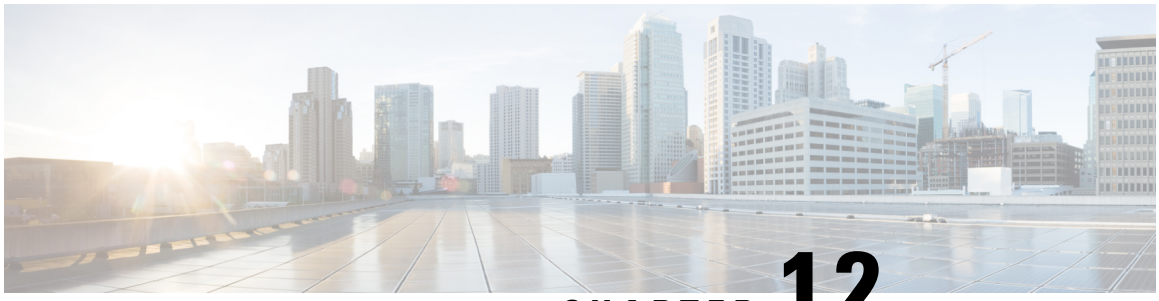
Feature History for COAP

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 Everest 16.6.1	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 17.7.1	COAP	This feature was implemented on supervisor modules C9400X-SUP-2 and C9400X-SUP-2XL, 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 http://www.cisco.com/go/cfn.



CHAPTER 12

Configuring an External USB Bluetooth Dongle

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)# bluetooth pin 1111
```

or

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

Note Cisco recommends using **bluetooth pin** command in global configuration mode.

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 Switch

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

Table 20: Commands to Monitor Bluetooth Settings on a Device

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

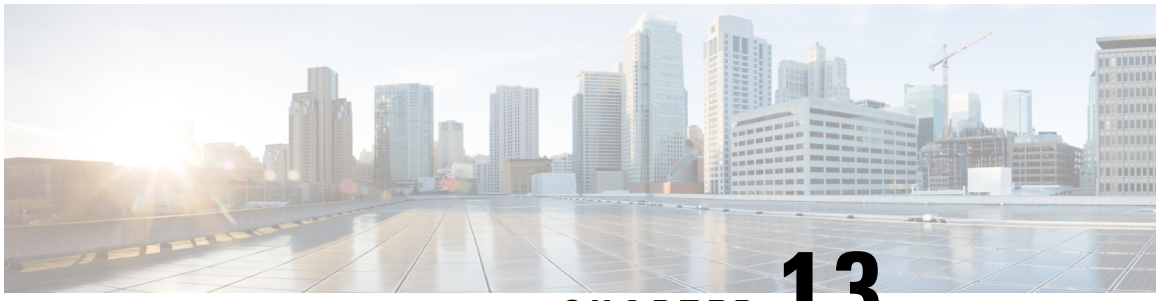
Feature History for Configuring an External Bluetooth Dongle

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 17.7.1	Configuring an External Bluetooth Dongle	This feature was implemented on supervisor modules C9400X-SUP-2 and C9400X-SUP-2XL, 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 http://www.cisco.com/go/cfn.



CHAPTER 13

M2 SATA Module

M2 SATA Module on Cisco Catalyst 9400 Series Supervisor

Cisco Catalyst 9400 is a next generation modular switch that lets you host applications for packet collection and analysis, testing, monitoring, and so on. To support the storage needs for these applications, the Cisco Catalyst 9400 Series Supervisor has an M2 connector that hosts a 22x88mm M2 SATA flash card. SATA configuration ranges from 240GB, 480GB to 960GB.

File System and Storage on M2 SATA

The default file system format of SATA is EXT4. However, SATA supports all extended file systems-EXT2, EXT3 and EXT4.

The SATA device has the following characteristics:

- Files stored on the M2 SATA partition are compatible with files stored on other devices.
- You can copy, or, store files between M2 SATA and other types of devices such as USB, eUSB, flash, and other IOS-XE file-system or storage.
- You can also read, write, delete, and format the SATA device.

.

Limitations of M2 SATA

- Non-EXT based file systems are not supported on M2 SATA.
- You cannot remove the M2 SATA device without powering off the Supervisor.
- You cannot use M2 SATA to boot images from ROMMON.
- You cannot upgrade the firmware on the M2 SATA drive.
- You cannot use M2 SATA to execute emergency install of images.

Self-Monitoring, Analysis and Reporting Technology System (S.M.A.R.T.) Health Monitoring

Cisco Catalyst IOS XE Release 16.9.1 gives you the ability to monitor the health of the device through CLIs. You can monitor internal hot-spots, flash wear-outs, and hardware failure of the SATA device and alert your users about a SATA failure. These users can then backup data and obtain a new SATA device.

A linux deamon smartd starts when the SATA is inserted into the Supervisor. By default, the polling interval is set to 2 days for offline test, 6 days for short test and 14 days for long test. The warnings and error messages are saved in /crashinfo/tracelogs/smart_errors.log and are also sent to the IOSd console.

The S.M.A.R.T. feature and smartd daemon are enabled by default when the SATA device is detected by the switch.

Note If the SATA is not detected after insertion, check the existing file system on the device. If it is not EXT based, SATA will not be detected. In that case, change the filesystem to EXT and reinsert the SATA.

The following CLI shows the logs from the smartd daemon:

```
Switch# more crashinfo:tracelogs/smart_errors.log
%IOSXEBOOT-4-SMART_LOG: (local/local): Mon Jan 4 00:13:10 Universal 2016
INFO: Starting SMART deamon
```

You can monitor the overall health of the device through the following CLI:

```
Switch# more flash:smart_overall_health.log
smartctl 6.4 2015-06-04 r4109 [x86_64-linux-4.4.131] (local build)
Copyright (C) 2002-15, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF READ SMART DATA SECTION ===
SMART overall-health self-assessment test result: PASSED
```

Accessing File System on M2 SATA

The mounted file system from the SATA flash card is accessed at disk0:. Use the **show file systems** command to view the details of each type of available filesystem.

Copying files to and from bootflash: or usbflash0: is supported.

Formatting the M2 SATA Flash Disk

To format a new Flash Disk, use the **format disk0:** command.

The format command recursively deletes all files on the device. This command fails if any file is open during its execution.

```
Switch#format disk0: ? <cr> <cr>
       ext2    ext2 filesystem type
       ext3    ext3 filesystem type
       ext4    ext4 filesystem type
       secure  Securely format the file system
<cr> <cr>


Switch# format disk0:
Format operation may take a while. Continue? [confirm]
Format operation will destroy all data in "disk0:". Continue? [confirm] Format of disk0:
complete
```

Operations on the SATA Module

The following are some of the operations that you can perform on the SATA:

Command	Description
dir *filesystem*	Displays the directories on the specified file system.
copy *source-file destination-url*	Copies files from specified source to a specified destination.
delete	Deletes a specified file
format	Formats the filesystem on the disk.
show disk0:	Displays the content and details of disk0:
show file information *file-url*	Displays information about a specific file.
show file systems	Displays the available file system on your device.
show inventory raw	Displays the details of the existing modules on the switch.

Following are sample outputs of the operations:

```
Switch# dir disk0:
Directory of disk0:/
        11  drwx            16384  May 11 2018 16:06:14 +00:00  lost+found
        10747905  drwx             4096  May 25 2018 13:03:43 +00:00  test
        236154740736 bytes total (224072925184 bytes free)
```

View the status of RP on a particular chassis:

```
Switch# dir disk0-1-1:
   Directory of disk0-1-1:/
```

```
   11  drwx            16384   Feb 1 2018 12:43:40 -08:00  lost+found
944994516992 bytes total (896892141568 bytes free)
```

Copy a file from the disk0: to USB

```
Switch# copy disk0:test.txt usbflash0:
Destination filename [test.txt]?
Copy in progress...C
17866 bytes copied in 0.096 secs (186104 bytes/sec)
```

```
Switch# dir usbflash0:
Directory of usbflash0:/
   12  -rw-         33554432  Jul 28 2017 10:12:58 +00:00  nvram_config
   11  drwx            16384  Jul 28 2017 10:09:46 +00:00  lost+found
   13  -rw-            17866  Aug 11 2017 09:52:16 +00:00  test.txt
189628416 bytes total (145387520 bytes free)
```

Delete the file test.txt from disk0:

```
Switch# delete disk0:test.txt
Delete filename [test.txt]?
Delete disk0:/test.txt? [confirm]
```

```
Switch# dir disk0:
Directory of disk0:/
No files in directory
118148280320 bytes total (112084135936 bytes free)
```

Copy file test.txt from USB to disk0:

```
Switch# copy usbflash0:test.txt disk0:
Destination filename [test.txt]?
Copy in progress...C
17866 bytes copied in 0.058 secs (308034 bytes/sec)

Switch# dir disk0:
Directory of disk0:/
   11  -rw-            17866  Aug 11 2017 09:53:03 +00:00  test.txt
118148280320 bytes total (112084115456 bytes free)
```

Format the disk

To format the ext4 filesystem, use the following command:

```
Switch#format disk0: ext4
```

Show commands

```
Switch# show disk0:
-#- --length-- ---------date/time--------- path
  2      17866 Aug 11 2017 09:54:06.0000000000 +00:00 test.txt
112084115456 bytes available (62513152 bytes used)
```

```
Switch# show file information disk0: test.txt
disk0:test.txt:
  type is image (elf64) []
  file size is 448 bytes, run size is 448 bytes
Foreign image, entry point 0x400610
```

```
Switch# show file systems
File Systems:

              Size(b)       Free(b)      Type  Flags  Prefixes
-
            -
*  11250098176    9694093312      disk     rw   bootflash: flash:
    1651314688    1232220160      disk     rw   crashinfo:
  118148280320  112084115456      disk     rw   disk0:
```

```
     189628416     145387520      disk     rw   usbflash0:
    7763918848    7696850944      disk     ro   webui:
             -             -    opaque     rw   null:
             -             -    opaque     ro   tar:
             -             -   network     rw   tftp:
      33554432      33532852     nvram     rw   nvram:
             -             -    opaque     wo   syslog:
             -             -   network     rw   rcp:
             -             -   network     rw   http:
             -             -   network     rw   ftp:
             -             -   network     rw   scp:
             -             -   network     rw   https:
             -             -    opaque     ro   cns:
```

```
Switch#show disk0: filesys
    Filesystem: disk0
    Filesystem Path: /vol/disk0
    Filesystem Type: ext4
    Mounted: Read/Write
```

```
Switch#show inventory raw
NAME: "Slot 5 SATA Container", DESCR: "SATA Container"
       PID:                   , VID:      , SN:
```

Feature History for M2 SATA Module

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 Everest 16.6.1	M2 SATA Module	The M2 SATA card addresses the storage needs of a device. It a a small form factor card and connector. For more information refer the *Hardware Installion Guide* for the device.
Cisco IOS XE Fuji 16.9.1	M2 SATA Module	Introduced support for application hosting storage needs.
Cisco IOS XE Cupertino 17.7.1	M2 SATA Module	This feature was implemented on supervisor modules C9400X-SUP-2 and C9400X-SUP-2XL, 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 http://www.cisco.com/go/cfn.