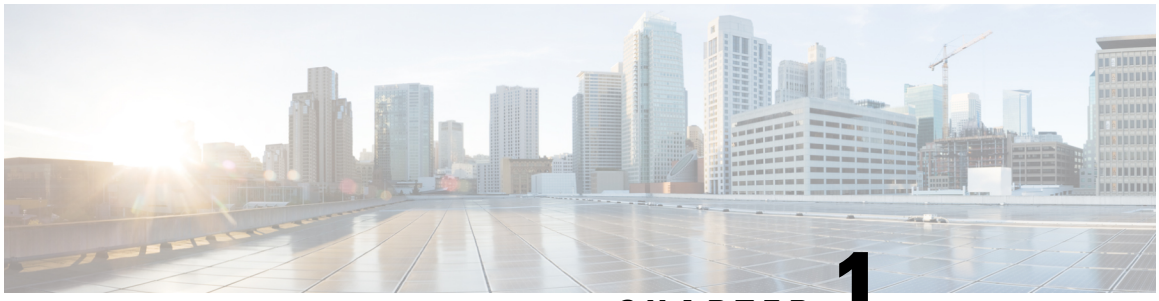
# Interface and Hardware Components Configuration Guide, Cisco IOS XE Bengaluru 17.4.x (Catalyst 9600 Switches)

**First Published:** 2020-11-30

**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. 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.

# Console Ports

## 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.

## 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.

```
switch-1

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

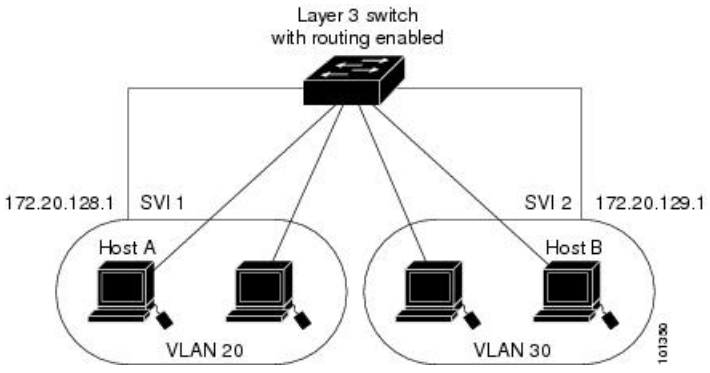
When the USB cable is removed or the PC de-activates the USB connection, the hardware automatically changes to the RJ-45 console interface:

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

# 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. An interface on the device is represented by a 3-tuple notation that lists the module, subslot, and port.

To configure a physical interface (port), specify the interface type, module number, sub-slot number, and device port number, and enter interface configuration mode.

- Type: 10-Gigabit Ethernet (TenGigabitEthernet or te) for 10 Gbps, 25-Gigabit Ethernet (TwentyFiveGigE or twe) for 25 Gbps, 40-Gigabit Ethernet (FortyGigabitEthernet or fo) for 40 Gbps, and 100-Gigabit Ethernet (HundredGigE or hu) for 100Gbps.
- Module number: The module or slot number on the device.
- Subslot number: The subslot number is always 0.
- Port number: The interface number on the device. The port numbering starts with the far left port when facing the front of the device, for example, FortyGigabitEthernet1/0/1.

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.

# Breakout Interfaces

Cisco Catalyst 9600 Series Switches support breakout cables. These cables support 4x10 G by enabling a single 40-G QSFP+ interface to be split into four 10-G SFP+ interfaces.

The default port connections depend on whether you use a 40-G QSFP module, a 100-G QSFP28 module, or a breakout cable. You can use a combination of 40-G QSFP modules, 100-G QSFP28 modules, and the 4x10 G breakout cables. Breakout interface naming can be as follows:

- HundredGigabitEthernet *slot-num*/0/*port-num*/[1-4]: For a device without Cisco StackWise Virtual (standalone device).
- HundredGigabitEthernet *switch-num*/*slot-num*/0/*port-num*/[1-4]: For a device with Cisco StackWise Virtual.

**Note** Breakout cable support is available only on the C9600-LC-24C line card, with a few limitations.

**Network Modules**

## Limitations for Breakout Interfaces

- Only the C9600-LC-24C line card supports breakout cables.
- Splitting a 100-G QSFP28 interface into four 25-G SFP28 interfaces is not supported.
- Breakout is supported only on 12 odd-numbered 100-G QSFP28 interfaces (top row ports) of the C9600-LC-24C line card.

  Physical port numbers 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21 and 23, that is, Hu1/0/25, Hu1/0/27, Hu1/0/29, Hu1/0/31, Hu1/0/33, Hu1/0/35, Hu1/0/37, Hu1/0/39, Hu1/0/41, Hu1/0/43, Hu1/0/45, and Hu1/0/47 support breakout.

  Physical port numbers 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22 and 24, that is, Hu1/0/26, Hu1/0/28, Hu1/0/30, Hu1/0/32, Hu1/0/34, Hu1/0/36, Hu1/0/38, Hu1/0/40, Hu1/0/42, Hu1/0/44, Hu1/0/46, and Hu1/0/48 do not support breakout.

# 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.

**Note** The interface is in Layer 2 by default.

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

***Table 1: Default Layer 2 Ethernet Interface Configuration***

| Feature | Default Setting |
|---|---|
| Operating mode | Layer 2 |
| Allowed VLAN range | VLANs 1– 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. |

| Feature | Default Setting |
|---|---|
| Speed | Speed is determined by the type of transceiver module plugged in. |
| Duplex mode | Full Duplex mode supported. |
| Flow control | Flow control is set to **receive: onsend: off**. |
| 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). |
| Port Fast | Disabled. |
| Auto-MDIX | Enabled. |

# Interface Speed and Duplex Mode

Switch modules include Ethernet (10/100/1000-Mbps) ports. The switch also includes multigigabit ethernet ports which support speeds up to 2.5 Gbps (100/1000/2500-Mbps), 5 Gbps (100/1000/2500/5000-Mbps), 10 Gbps (100/1000/2500/5000/10000-Mbps); SFP modules that support speeds up to 1 Gbps, SFP+ modules that support speeds up to 10 Gbps, SFP28 modules that support speeds up to 25 Gbps, QSFP modules that support speeds up to 40 Gbps and 100 Gbps.

In full-duplex mode, two stations can send and receive traffic at the same time.

# Speed and Duplex Configuration Guidelines

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

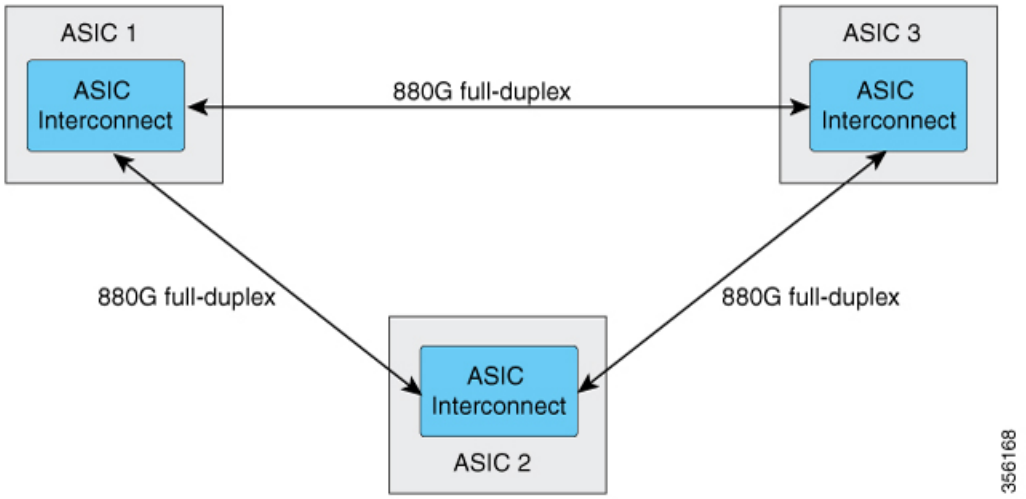
- Ethernet ports operating at 10Mbps/100Mbps/1Gbps/2.5Gbps/5/Gbps/10Gbps support full duplex mode. Half duplex mode is not supported.
- If both ends of the line support autonegotiation, we highly recommend the default setting of **auto** negotiation.
- If one interface supports autonegotiation and the 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.

# Port Mapping and Oversubscription

The Catalyst 9600 modular chassis supports up to four line cards and redundant supervisors. The supervisor has three Unified Access Data Path (UADP 3.0) ASICs connected to each other. Each UADP 3.0 ASIC provides a switching capacity of 1600Gbps full-duplex for the front panel interfaces, resulting in a total capacity of 4800 Gbps full-duplex switching capacity.

***Figure 2: Three-ASIC Configuration***




Each ASIC also provides a total of 1760 Gbps full-duplex inter-ASIC bandwidth, with 880 Gbps full-duplex to each of the other two ASICs.

The inter-ASIC connections use a broadcast network approach, to ensure that user data is available to all ASICs. In the worst-case scenario, where all traffic is inter-ASIC, the front panel bandwidth can be 2:1 oversubscribed compared to the interface-ASIC bandwidth. Most traffic scenarios (example north-to-south) will only require some traffic to be inter-ASIC.

When a Line card is installed on the chassis, one-third of its ports are connected to each ASIC. Which means, one set of ports on the line card are connected to ASIC 1, the second set to ASIC 2, and the third set of ports are connected to ASIC 3. You can view the port mapping on the line cards using the **show platform software fed active ifm mapping** command.

## Port Mapping on C9600-LC-24C

By default, all the interfaces on a C9600-LC-24C line card are 40 G or 1 G enabled. You can configure the odd-numbered 40 G or 1 G interface to function as 100 G port using the **enable** command on the interface. In such a case, the corresponding even-numbered port in the port group is disabled. (A port group constitutes the top and bottom consecutive ports.)

*Figure 3: Port Numbering on C9600-LC-24C*

| Both 40G/100G ports | 1/25 | 3/27 | 5/29 | 7/31 | 9/33 | 11/35 | 13/37 | 15/39 | 17/41 | 19/43 | 21/45 | 23/47 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Only 40G ports | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 | 22 | 24 |

*Figure 4: Default Configuration on C9600-LC-24C*

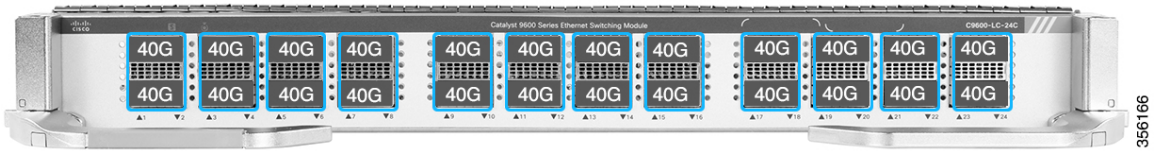
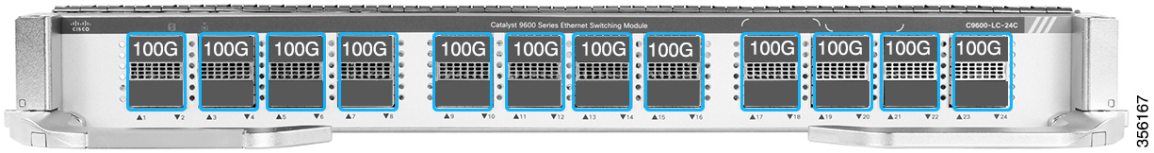



*Figure 5: 100 G Configuration on C9600-LC-24C*




C9600-LC-24C supports CVR-QSFP-SFP10G (QSA adapter) that provides 10 G connectivity on QSFP ports by converting a 40 G or 100 G QSFP port into an SFP/SFP+ port.

C9600-LC-24C supports only the following port group configurations with CVR-QSFP-SFP10G:

- Configuring odd-numbered (top) and even-numbered (bottom) ports with the QSA adapter
- Configuring odd-numbered ports with the QSA adapter and even-numbered ports with 40 G QSFP optics

**Note** In a port group, if you configure the odd-numbered port with 40 G QSFP optics and the even-numbered port with the QSA adapter, the QSA adapter in the even-numbered port doesn't work.

# IEEE 802.3x Flow Control

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

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

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

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

These rules apply to flow control settings on the device:

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

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

# Layer 3 Interfaces

The device supports these types of Layer 3 interfaces:

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

**Note**
- When you create an SVI, it does not become active until it is associated with a physical port.
- SVI MAC addresses do not change after a device reload. This is expected behavior.

  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.

You can configure a maximum of 4000 Layer 3 interfaces. 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**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>Enter your password, if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **interface**<br>**Example:**<br>`Device(config)# interface fortygigabitethernet1/0/1`<br>`Device(config-if)#` | Identifies the interface type, and the number of the connector.<br>**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 **fortygigabitethernet 1/0/1**, or **fortygigabitethernet1/0/1**. |
| **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.<br>**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**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>Enter your password, if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **interface** *interface-id*<br>**Example:**<br>`Device(config)# interface fortygigabitethernet1/0/2` | Specifies the interface for which you are adding a description, and enter interface configuration mode. |
| **Step 4** | **description** *string*<br>**Example:**<br>`Device(config-if)# description Connects to Marketing` | Adds a description for an interface. |
| **Step 5** | **end**<br>**Example:**<br>`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**<br>**Example:**<br>`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**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>Enter your password, if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **interface range** {*port-range* \| **macro** *macro_name*}<br>**Example:**<br>`Device(config)# interface range macro` | Specifies the range of interfaces (VLANs or physical ports) to be configured, and enter interface-range configuration mode.<br>• You can use the **interface range** command to configure up to five port ranges or a previously defined macro.<br>• The **macro** variable is explained in Configuring and Using Interface Range Macros.<br>• In a comma-separated *port-range*, you must enter the interface type for each entry and enter spaces before and after the comma.<br>• 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.<br>**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**<br>**Example:**<br>`Device(config)# end` | Returns to privileged EXEC mode. |
| **Step 5** | **show interfaces** [*interface-id*]<br>**Example:**<br>`Device# show interfaces` | Verifies the configuration of the interfaces in the range. |
| **Step 6** | **copy running-config startup-config**<br>**Example:**<br>`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**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>Enter your password, if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **define interface-range** *macro_name interface-range*<br>**Example:** | Defines the interface-range macro, and saves it in NVRAM.<br>• The *macro_name* is a 32-character maximum character string.<br>• A macro can contain up to five comma-separated interface ranges.<br>• Each *interface-range* must consist of the same port type.<br>**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. |
| **Step 4** | **interface range macro** *macro_name*<br>**Example:**<br>`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*.<br>You can now use the normal configuration commands to apply the configuration to all interfaces in the defined macro. |
| **Step 5** | **end**<br>**Example:** | Returns to privileged EXEC mode. |

| | Command or Action | Purpose |
|---|---|---|
| | `Device(config)# end` | |
| **Step 6** | **show running-config \| include define**<br>**Example:**<br>`Device# show running-config \| include define` | Shows the defined interface range macro configuration. |
| **Step 7** | **copy running-config startup-config**<br>**Example:**<br>`Device# copy running-config startup-config` | (Optional) Saves your entries in the configuration file. |

# Setting the Interface Speed and Duplex Parameters

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

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>Enter your password, if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface** *interface-id*<br>**Example:**<br>`Device(config)# interface fortygigabitethernet1/0/3` | Specifies the physical interface to be configured, and enters interface configuration mode. |
| **Step 4** | **speed** {**10** \| **100** \| **1000** \| **10000** \| **2500** \| **5000** \| **auto** [**10** \| **100** \| **1000** \| **10000** \| **2500** \| **5000**]}<br>**Example:**<br>`Device(config-if)# speed 10` | Enters the appropriate speed parameter for the interface:<br>• Enter **10**, **100**, **1000**, **10000**, **2500**, or **5000** to set a specific speed for the interface. |

| | Command or Action | Purpose |
|---|---|---|
| | | • 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. |
| **Step 5** | **duplex** {**auto** \| **full**}<br>**Example:**<br>`Device(config-if)# duplex full` | Enters the duplex parameter for the interface.<br>You can configure the duplex setting when the speed is set to **auto**. |
| **Step 6** | **end**<br>**Example:**<br>`Device(config-if)# end` | Returns to privileged EXEC mode. |
| **Step 7** | **show interfaces** *interface-id*<br>**Example:**<br>`Device# show interfaces fortygigabitethernet1/0/3` | Displays the interface speed and duplex mode configuration. |
| **Step 8** | **copy running-config startup-config**<br>**Example:**<br>`Device# copy running-config startup-config` | (Optional) Saves your entries in the configuration file. |

# Configuring a Breakout Interface

Follow these steps to configure a breakout interface.

### Procedure

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

| | Command or Action | Purpose |
|---|---|---|
| | `Device# configure terminal` | |
| **Step 3** | **hw-module breakout-enable**<br>**Example:**<br>`Device(config)# hw-module breakout-enable` | Enables the Breakout feature. |
| **Step 4** | **exit**<br>**Example:**<br>`Device(config)# exit` | Exits global configuration mode. |
| **Step 5** | **reload**<br>**Example:**<br>`Device# reload` | Reloads the system.<br>After the system has reloaded, enter the global configuration mode and perform the steps below to configure breakout interfaces. |
| **Step 6** | **interface** *interface-id*<br>**Example:**<br>`Device(config)# interface HundredGigabitEthernet1/0/25` | Specifies the interface to be configured, and enters interface configuration mode. |
| **Step 7** | **enable**<br>**Example:**<br>`Device(config-if)# enable` | Enables breakout on an interface. |
| **Step 8** | **exit**<br>**Example:**<br>`Device(config-if)# exit` | Exits interface configuration mode and returns to global configuration mode. |
| **Step 9** | Run one of the following commands:<br>• **hw-module slot** *slot-num* **breakout** *port-num*<br>• **hw-module switch** *switch-num* **slot** *slot-num* **breakout** *port-num*<br>**Example:** | Enables breakout on the specified port.<br>• Use this command on a device without Cisco StackWise Virtual (standalone mode).<br>• Use this command on a device with Cisco StackWise Virtual. |

| | Command or Action | Purpose |
|---|---|---|
| | `Device(config)# hw-module slot 1 breakout 25`<br>**Example:**<br>`Device(config)# hw-module switch 1 slot 1 breakout 25` | **Note** To enable breakout on a range of ports, use the **breakout range** *port-range* command. |
| **Step 10** | **end**<br>**Example:**<br>`Device(config)# end` | Exits global configuration mode and returns to privileged EXEC mode. |
| **Step 11** | **show interface status**<br>**Example:**<br>`Device# show interface status` | (Optional) Verifies the configuration. |

# Configuring Hundred Gigabit Ethernet Interface on C9600-LC-24C

By default, 40G is enabled on all ports of C9600-LC-24C linecard. You can enable 100 G on the odd-numbered ports (ports 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, and 47) of the linecard. When 100 G is enabled, 40 G is disabled on the corresponding port and the port below it.

Follow these steps to enable 100 G on the ports:

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface** *interface-type interface-id*<br>**Example:**<br>`Device(config)# interface HundredGigabitEthernet1/0/27` | Specifies the interface that is to be configured. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | **enable**<br>**Example:**<br>`Device(config-if)# enable` | Enables the hundred gigabit ethernet interface. Use **no enable** command to disable the hundred gigabit ethernet interface. |

# 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**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>Enter your password, if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode |
| **Step 3** | **interface** *interface-id*<br>**Example:**<br>`Device(config)# interface fortygigabitethernet1/0/1` | Specifies the physical interface to be configured, and enters interface configuration mode. |
| **Step 4** | **flowcontrol** {**receive**} {**on** \| **off** \| **desired**}<br>**Example:**<br>`Device(config-if)# flowcontrol receive on` | Configures the flow control mode for the port. |
| **Step 5** | **end**<br>**Example:**<br>`Device(config-if)# end` | Returns to privileged EXEC mode. |
| **Step 6** | **show interfaces** *interface-id*<br>**Example:**<br>`Device# show interfaces` | Verifies the interface flow control settings. |

| | Command or Action | Purpose |
|---|---|---|
| | `fortygigabitethernet1/0/1` | |
| **Step 7** | **copy running-config startup-config**<br>**Example:**<br>`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**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>Enter your password, if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface** {**fortygigabitethernet** *interface-id*} \| {**vlan** *vlan-id*} \| {**port-channel** *port-channel-number*}<br>**Example:**<br>`Device(config)# interface fortygigabitethernet1/0/2` | Specifies the interface to be configured as a Layer 3 interface, and enters interface configuration mode. |
| **Step 4** | **no switchport**<br>**Example:**<br>`Device(config-if)# no switchport` | (For physical ports only) Enters Layer 3 mode. |
| **Step 5** | **ip address** *ip_address subnet_mask*<br>**Example:**<br>`Device(config-if)# ip address` | Configures the IP address and IP subnet. |

| | Command or Action | Purpose |
|---|---|---|
| | `192.20.135.21 255.255.255.0` | |
| **Step 6** | **no shutdown**<br>**Example:**<br>`Device(config-if)# no shutdown` | Enables the interface. |
| **Step 7** | **end**<br>**Example:**<br>`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**<br>**Example:**<br>`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 900 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**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>Enter your password, if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface tunnel** *number*<br>**Example:**<br>`Device(config)#interface tunnel 2` | Enables tunneling on the interface. |
| **Step 4** | **ip address** *ip_addresssubnet_mask*<br>**Example:**<br>`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*}<br>**Example:**<br>`Device(config)#tunnel source 10.10.10.1` | Configures the tunnel source. |
| **Step 6** | **tunnel destination** {*host_name* \| *ip_address*}<br>**Example:**<br>`Device(config)#tunnel destination 10.10.10.2` | Configures the tunnel destination. |
| **Step 7** | **tunnel mode gre ip**<br>**Example:**<br>`Device(config)#tunnel mode gre ip` | Configures the tunnel mode. |
| **Step 8** | **end**<br>**Example:**<br>`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**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>Enter your password, if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface** *interface-id*<br>**Example:**<br>`Device(config)# interface fortygigabitethernet1/0/2` | Specifies a Layer 2 interface (physical port or port channel), and enters interface configuration mode. |
| **Step 4** | **switchport autostate exclude**<br>**Example:**<br>`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**<br>**Example:**<br>`Device(config-if)# end` | Returns to privileged EXEC mode. |
| **Step 6** | **show running config interface** *interface-id* | (Optional) Shows the running configuration.<br>Verifies the configuration. |
| **Step 7** | **copy running-config startup-config**<br>**Example:**<br>`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**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>Enter your password, if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface** {**vlan** *vlan-id*} \| { **fortygigabitethernet** *interface-id*} \| {**port-channel** *port-channel-number*}<br>**Example:**<br>`Device(config)# interface fortygigabitethernet1/0/2` | Selects the interface to be configured. |
| **Step 4** | **shutdown**<br>**Example:**<br>`Device(config-if)# shutdown` | Shuts down an interface. |
| **Step 5** | **no shutdown**<br>**Example:**<br>`Device(config-if)# no shutdown` | Restarts an interface. |
| **Step 6** | **end**<br>**Example:**<br>`Device(config-if)# end` | Returns to privileged EXEC mode. |
| **Step 7** | **show running-config**<br>**Example:**<br>`Device# show running-config` | Verifies your entries. |

# Configuring the Console Media Type

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

**Procedure**

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

# 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**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>Enter your password, if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **line console 0**<br>**Example:**<br>`Device(config)# line console 0` | Configures the console and enters line configuration mode. |
| **Step 4** | **usb-inactivity-timeout switch** *switch_number timeout-minutes*<br>**Example:**<br>`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**<br>**Example:**<br>`Device# copy running-config startup-config` | (Optional) Saves your entries in the configuration file. |

# Monitoring Interface Characteristics

The following sections provide information about monitoring interface characteristics.

## Monitoring Interface Status

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

***Table 2: show Commands for Interfaces***

| Command | Purpose |
|---|---|
| **show interfaces** *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. |
| **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 3: clear Commands for Interfaces***

| Command | Purpose |
|---|---|
| **clear counters** [*interface-id*] | Clears interface counters. |

| Command | Purpose |
|---|---|
| **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 fortygigabitethernet1/0/2
Device(config-if)# description Connects to Marketing
Device(config-if)# end
Device# show interfaces fortygigabitethernet1/0/2 description
Interface Status    Protocol Description
Fo1/0/1   down      down      Connects to Marketing
```

## Example: Configuring a Range of Interfaces

The following example shows how to use the **interface range** global configuration command to shut down ports 1 to 2 on switch 1:

```
Device# configure terminal
Device(config)# interface range fortyGigabitEthernet 1/0/1-2
Device(config-if-range)# shut
```

**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 fortygigabitethernet1/0/3
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 fortygigabitethernet1/0/2
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 fortygigabitethernet1/0/2
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 a Breakout Interface

The following example shows a sample output of the **show interface status** command for a specified interface:

```
Device# show interface status | include 1/0/25

Hu1/0/25            inactive     1            full    40G unknown
Hu1/0/25/1          connected    101          full    10G QSFP 4X10G AC10M SFP
Hu1/0/25/2          connected    102          full    10G QSFP 4X10G AC10M SFP
Hu1/0/25/3          connected    103          full    10G QSFP 4X10G AC10M SFP
Hu1/0/25/4          connected    104          full    10G QSFP 4X10G AC10M SFP
```

The following example shows a sample output of the running configuration for the Breakout feature on the device:

```
Device# show running-config | include breakout

hw-module  slot 1 breakout 25
hw-module breakout-enable
```

# Example: Configuring the Console Media Type

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

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

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

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

# Example: Configuring USB Inactivity Timeout

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

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

The following example shows how to disable the configuration:

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

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

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

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

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

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

# Additional References for Configuring Interface Characteristics

### Related Documents

| Related Topic | Document Title |
|---|---|
| For complete syntax and usage information for the commands used in this chapter. | See the "Interface and Hardware Commands" section in the *Command Reference (Catalyst 9600 Series Switches)* . |

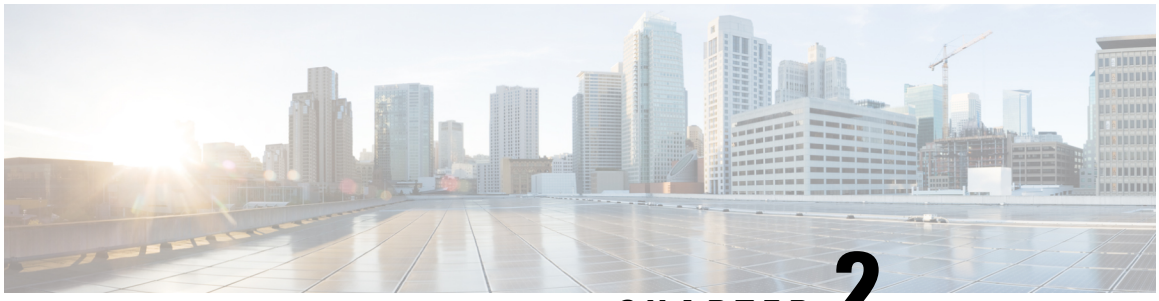
# Feature History for Configuring Interface Characteristics

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 Gibraltar 16.11.1 | 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 Amsterdam 17.1.1 | Ethernet and Multi-Gigabit Ethernet Interfaces | Support for Ethernet and Multi-Gigabit Ethernet ports operating at 10 Mbps, 100 Mbps, 1000 Mbps, 2.5 Gbps, 5 Gbps, and 10 Gbps was introduced on all models of the series. |
| Cisco IOS XE Amsterdam 17.2.1 | Breakout interfaces | Support for breakout configuration was introduced only on the 12 ports of the top row (odd numbers) of C9600-LC-24C line card. |

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 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.

In addition, Cisco Catalyst 9600 Series Switches have another Ethernet management port, *TenGigabitEthernet0/1*, an SFP+ interface that provides pluggable connectivity from the device to the management network. This interface supports 10G and 1G Transceivers.

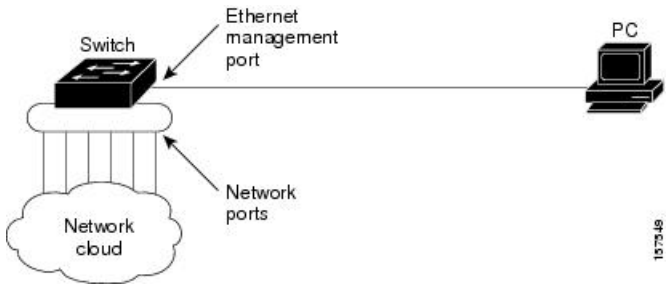
By default, *GigabitEthernet0/0* is enabled. You can disable *GigabitEthernet0/0* and enable *TenGigabitEthernet0/1* by doing either one of the following:

- Using the **platform management-interface TenGigabitEthernet0/1** command and *rebooting* the device. You can switch back to the default management port by using the **no platform management-interface TenGigabitEthernet0/1** command and *rebooting* the device.
- Setting the environment variable ETHER_PORT to 1 in the ROMMON mode and *rebooting* the switch. You can switch back to the default management port by setting the ETHER_PORT variable to 2 in ROMMON mode and *rebooting* the switch.

# Ethernet Management Port Direct Connection to a Device

*Figure 6: 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.

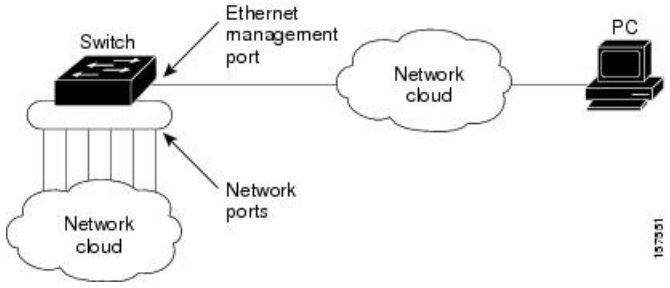
**Note** Any SSH, SCP, or Telnet sessions established by clients over the Ethernet management port IP address before stateful switchover to a new active switch in StackWise Virtual will be terminated and a new session has to be initiated after switchover.

# 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 7: 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 (default)
  - 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**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 2** | **interface gigabitethernet0/0**<br>**Example:**<br>`Device(config)# interface gigabitethernet0/0` | Specifies the Ethernet management port in the CLI. |
| **Step 3** | **shutdown**<br>**Example:**<br>`Device(config-if)# shutdown` | Disables the Ethernet management port. |
| **Step 4** | **no shutdown**<br>**Example:**<br>`Device(config-if)# no shutdown` | Enables the Ethernet management port. |
| **Step 5** | **exit**<br>**Example:**<br>`Device(config-if)# exit` | Exits interface configuration mode. |
| **Step 6** | **show interfaces gigabitethernet0/0**<br>**Example:**<br>`Device# show interfaces gigabitethernet0/0` | Displays the link status.<br>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.

# Enabling TenGigabitEthernet Management Port

Follow these steps to enable the SFP+ interface that provides pluggable connectivity from the device to the management interface. Enabling the TenGigabitEthernet management port on the supervisor disables the GigabitEthernet management port.

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 2** | **platform management-interface TenGigabitEthernet0/1**<br>**Example:**<br>`Device(config)# platform management-interface TenGigabitEthernet0/1` | Changes the default management port of GigabitEthernet0/0 to TenGigabitEthernet0/1.<br>This change will take effect after rebooting the device.<br>To switch back to the default management port, use the **no** form of this command, and reboot the device. |
| **Step 3** | **end**<br>**Example:**<br>`Device(config)# end` | Exits the gobal configuration mode and enters privileged EXEC mode. |
| **Step 4** | **show platform management-interface**<br>**Example:**<br>`Device# show platform management-interface` | (Optional) Displays information about the active and configured management port. |
| **Step 5** | **reload**<br>**Example:**<br>`Device# reload` | Boots up the device. |
| **Step 6** | **show platform management-interface**<br>**Example:**<br>`Device# show platform management-interface` | (Optional) Displays information about the active management port. |

# 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
```

# Monitoring the Ethernet Management Port

Commands entered at the privileged EXEC prompt display information about the management port, including the list of transceivers that are supported on the pluggable management port.

***Table 4: show Commands for Ethernet Management Port***

| Command | Purpose |
|---|---|
| **show platform management-interface** | Displays the active management port. |
| **show interfaces transceiver supported-list \| b management interface** | Displays the list of transceivers that are supported on the pluggable management port. |

The following example shows a sample output of the **show platform management-interface** command. The command output displays the active management port.

```
Device# show platform management-interface

Management interface is GigabitEthernet0/0
```

The following example shows a sample output of the **show interfaces transceiver supported-list | b management interface** command. The command output displays all the transceivers that are supported on the pluggable management port.

```
Device# show interfaces transceiver supported-list | b management interface

Transceivers supported on management interface TenGigabitEthernet0/1:
   GLC-SX-MM                NONE
   GLC-SX-MMD               ALL
   SFP-10G-LR               ALL
   SFP-10G-LR-S             ALL
   SFP-10G-SR               ALL
   SFP-10G-SR-S             ALL
   SFP-H10GB-CU1M           NONE
   SFP-H10GB-CU3M           NONE
   SFP-H10GB-CU5M           NONE
```

# 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 (Catalyst 9600 Series Switches)* |

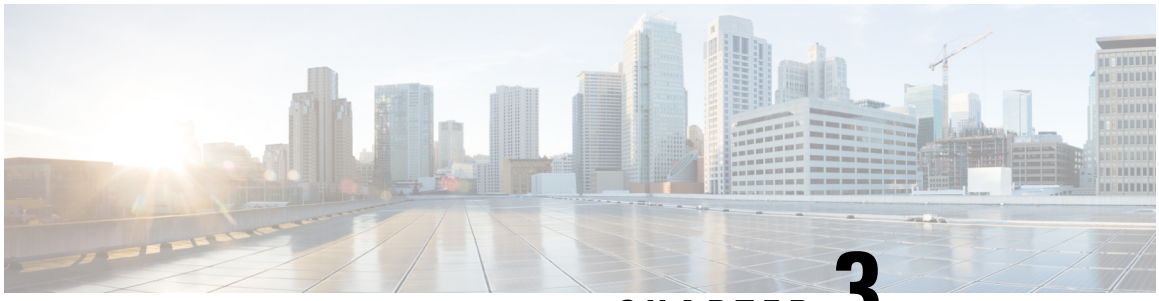
# Feature History for Ethernet Management Port

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 Gibraltar 16.11.1 | 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 Gibraltar 16.12.x | Changing Ethernet Management Port | The **platform management-interface** command was introduced to change the default management port. |

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

# Checking Port Status and Connectivity



# Check Cable Status Using Time Domain Reflectometer

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

## Running the TDR Test

To start the TDR test, perform this task:

**Procedure**

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

## TDR Guidelines

The following guidelines apply to the use of TDR:

- Do not change the port configuration while the TDR test is running.
- If you connect a port undergoing a TDR test to an Auto-MDIX enabled port, the TDR result might be invalid.
- If you connect a port undergoing a TDR test to a 100BASE-T port such as that on the device, the unused pairs (4-5 and 7-8) are reported as faulty because the remote end does not terminate these pairs.
- Due to cable characteristics, you should run the TDR test multiple times to get accurate results.

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

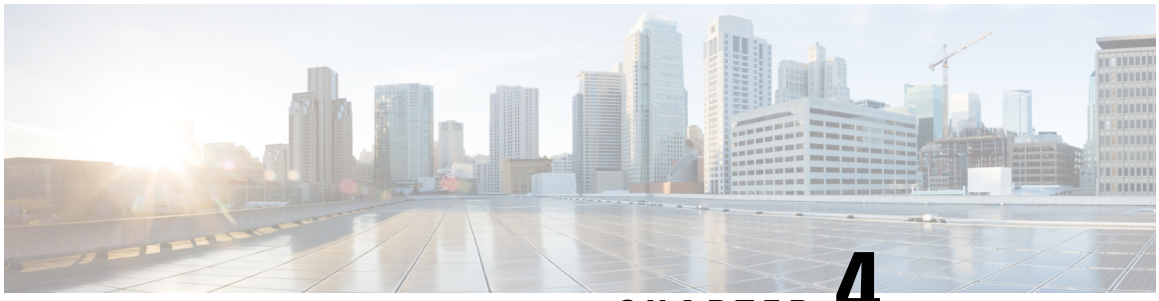
# Feature History for Checking Port Status and Connectivity

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

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

| Release | Feature | Feature Information |
|---|---|---|
| Cisco IOS XE Gibraltar 16.11.1 | Time Domain Reflectometer (TDR) | TDR allows you to determine if a cable is OPEN or SHORT when it is at fault. |

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

# 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*] } interface configuration command. By default the PoE interface is in **auto** mode;

- Inventory management TLV

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

- Location TLV

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

  - Civic location information

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

  - ELIN location information

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

  - Geographic location information

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

  - custom location

    Provides customized name and value of a switch location.

# Wired Location Service

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

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

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

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

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

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

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

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

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

# Default LLDP Configuration

*Table 5: 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**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **lldp run**<br>**Example:**<br>`Device(config)# lldp run` | Enables LLDP globally on the device. |
| **Step 4** | **interface** *interface-id*<br>**Example:**<br>`Device(config)# interface gigabitethernet2/0/1` | Specifies the interface on which you are enabling LLDP, and enter interface configuration mode. |
| **Step 5** | **lldp transmit**<br>**Example:**<br>`Device(config-if)# lldp transmit` | Enables the interface to send LLDP packets. |
| **Step 6** | **lldp receive**<br>**Example:**<br>`Device(config-if)# lldp receive` | Enables the interface to receive LLDP packets. |
| **Step 7** | **end**<br>**Example:**<br>`Device(config-if)# end` | Returns to privileged EXEC mode. |
| **Step 8** | **show lldp**<br>**Example:**<br>`Device# show lldp` | Verifies the configuration. |
| **Step 9** | **copy running-config startup-config**<br>**Example:**<br>`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**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **lldp holdtime** *seconds*<br>**Example:**<br>`Device(config)# lldp holdtime 120` | (Optional) Specifies the amount of time a receiving device should hold the information from your device before discarding it.<br>The range is 0 to 65535 seconds; the default is 120 seconds. |
| **Step 4** | **lldp reinit** *delay*<br>**Example:**<br>`Device(config)# lldp reinit 2` | (Optional) Specifies the delay time in seconds for LLDP to initialize on an interface.<br>The range is 2 to 5 seconds; the default is 2 seconds. |
| **Step 5** | **lldp timer** *rate*<br>**Example:**<br>`Device(config)# lldp timer 30` | (Optional) Sets the sending frequency of LLDP updates in seconds.<br>The range is 5 to 65534 seconds; the default is 30 seconds. |
| **Step 6** | **lldp tlv-select**<br>**Example:**<br>`Device(config)# tlv-select` | (Optional) Specifies the LLDP TLVs to send or receive. |
| **Step 7** | **interface** *interface-id*<br>**Example:**<br>`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**<br>**Example:**<br>`Device(config-if)# lldp med-tlv-select inventory management` | (Optional) Specifies the LLDP-MED TLVs to send or receive. |
| **Step 9** | **end**<br>**Example:**<br>`Device(config-if)# end` | Returns to privileged EXEC mode. |
| **Step 10** | **show lldp**<br>**Example:**<br>`Device# show lldp` | Verifies the configuration. |
| **Step 11** | **copy running-config startup-config**<br>**Example:**<br>`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 6: 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**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface** *interface-id*<br>**Example:**<br>`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**<br>**Example:**<br>`Device(config-if)# lldp med-tlv-select inventory management` | Specifies the TLV to enable. |
| **Step 5** | **end**<br>**Example:**<br>`Device(config-if)# end` | Returns to privileged EXEC mode. |
| **Step 6** | **copy running-config startup-config**<br>**Example:**<br>`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:**<br>`Device> enable` | • Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **network-policy profile** *profile number*<br>**Example:**<br>`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**]<br>**Example:**<br>`Device(config-network-policy)# voice vlan 100 cos 4` | Configures the policy attributes:<br>• **voice**—Specifies the voice application type.<br>• **voice-signaling**—Specifies the voice-signaling application type.<br>• **vlan**—Specifies the native VLAN for voice traffic.<br>• *vlan-id*—(Optional) Specifies the VLAN for voice traffic. The range is 1 to 4094.<br>• **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.<br>• **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.<br>• **dot1p**—(Optional) Configures the telephone to use IEEE 802.1p priority tagging and use VLAN 0 (the native VLAN).<br>• **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**<br>**Example:**<br>`Device(config)# exit` | Returns to global configuration mode. |
| **Step 6** | **interface** *interface-id*<br>**Example:**<br>`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*<br>**Example:**<br>`Device(config-if)# network-policy 1` | Specifies the network-policy profile number. |
| **Step 8** | **lldp med-tlv-select network-policy**<br>**Example:**<br>`Device(config-if)# lldp med-tlv-select network-policy` | Specifies the network-policy TLV. |
| **Step 9** | **end**<br>**Example:**<br>`Device(config)# end` | Returns to privileged EXEC mode. |
| **Step 10** | **show network-policy profile**<br>**Example:**<br>`Device# show network-policy profile` | Verifies the configuration. |
| **Step 11** | **copy running-config startup-config**<br>**Example:**<br>`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**<br>**Example:**<br>`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**}}<br>**Example:**<br>`Device(config)# location civic-location identifier 1`<br>`Device(config-civic)# number 3550`<br>`Device(config-civic)# primary-road-name "Cisco Way"`<br>`Device(config-civic)# city "San Jose"`<br>`Device(config-civic)# state CA`<br>`Device(config-civic)# building 19`<br>`Device(config-civic)# room C6`<br>`Device(config-civic)# county "Santa Clara"`<br>`Device(config-civic)# country US` | Specifies the location information for an endpoint.<br>• **admin-tag**—Specifies an administrative tag or site information.<br>• **civic-location**—Specifies civic location information.<br>• **elin-location**—Specifies emergency location information (ELIN).<br>• **custom-location**—Specifies custom location information.<br>• **geo-location**—Specifies geo-spatial location information.<br>• **identifier** *id*—Specifies the ID for the civic, ELIN, custom, or geo location.<br>• **host**—Specifies the host civic, custom, or geo location.<br>• *string*—Specifies the site or location information in alphanumeric format. |
| **Step 3** | **exit**<br>**Example:**<br>`Device(config-civic)# exit` | Returns to global configuration mode. |
| **Step 4** | **interface** *interface-id*<br>**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**} }<br>**Example:**<br>`Device(config-if)# location elin-location-id 1` | • **additional-location-information**—Specifies additional information for a location or place.<br>• **civic-location-id**—Specifies global civic location information for an interface.<br>• **elin-location-id**—Specifies emergency location information for an interface.<br>• **custom-location-id**—Specifies custom location information for an interface.<br>• **geo-location-id**—Specifies geo-spatial location information for an interface.<br>• **host**—Specifies the host location identifier.<br>• *word*—Specifies a word or phrase with additional location information.<br>• *id*—Specifies the ID for the civic, ELIN, custom, or geo location. The ID range is 1 to 4095. |
| **Step 6** | **end**<br>**Example:**<br>`Device(config-if)# end` | Returns to privileged EXEC mode. |
| **Step 7** | Use one of the following:<br>• **show location admin-tag** *string*<br>• **show location civic-location identifier** *id*<br>• **show location elin-location identifier** *id*<br>**Example:**<br>`Device# show location admin-tag`<br>or<br>`Device# show location civic-location identifier`<br>or<br>`Device# show location elin-location identifier` | Verifies the configuration. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 8** | **copy running-config startup-config**<br>**Example:**<br>`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**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **nmsp notification interval** {**attachment** \| **location**} *interval-seconds*<br>**Example:**<br>`Device(config)# nmsp notification interval location 10` | Specifies the NMSP notification interval.<br>**attachment**—Specifies the attachment notification interval.<br>**location**—Specifies the location notification interval.<br>*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**<br>**Example:**<br>`Device(config)# end` | Returns to privileged EXEC mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | **show network-policy profile**<br>**Example:**<br>`Device# show network-policy profile` | Verifies the configuration. |
| **Step 6** | **copy running-config startup-config**<br>**Example:**<br>`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.<br>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.<br>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.<br>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 9600 Series Switches)* |

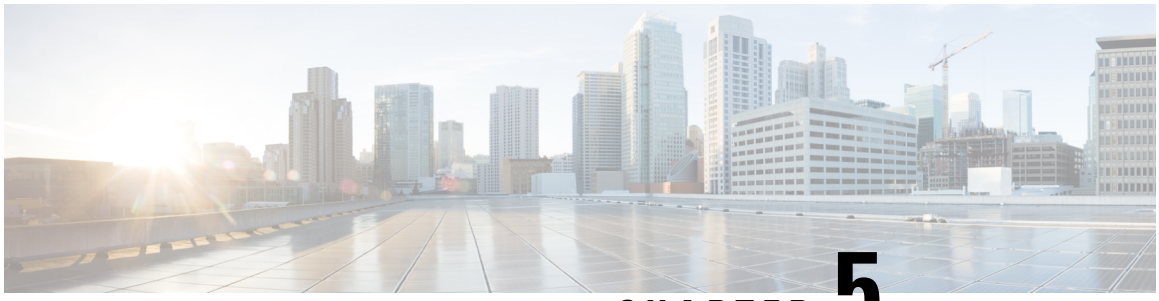
# Feature History for LLDP, LLDP-MED, and Wired Location Service

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 Gibraltar 16.11.1 | 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.<br>LLDP-MED operates between endpoints and network devices.<br>Wired Location Service lets you send tracking information of the connected devices to a Cisco Mobility Services Engine (MSE). |

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 Link Debounce Timer



## Restrictions for Link Debounce Timer

- Link Debounce Timer is not available on StackWise Virtual Links (SVL). It is available only on non-SVL links.

## Information About Link Debounce Timer

The Link Debounce Timer delays notification of a link down status change. Delayed notification of a link down status change can decrease traffic loss due to network reconfiguration when network ethernet port experiences minor faults in the link.

If the status of a link changes quickly from up to down and then back to up, the Link Debounce Timer suppresses the link down status notification. If the link transitions from up to down, but does not come back up, the Link Debounce Timer delays the link down status notification until the debounce timer expires.

Delayed link down status notification allows a quick port status change and recovery without triggering any of the changes that are necessary when a port goes down. The normal operation of Dense Wavelength Division Multiplexing (DWDM) links includes quick port status changes and recovery during DWDM network reconvergence. Delayed link status notification can also be used to mitigate link flaps because of bad cabling.

You can configure the port debounce timer separately on each LAN port.

**Note** Enabling the port debounce timer causes link down detections to be delayed, resulting in loss of traffic during the debouncing period. This situation might affect the convergence and reconvergence of some Layer 2 and Layer 3 protocols.

# Configuring Link Debounce Timer

To configure Link Debounce Timer on a port, perform this task:

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode. Enter your password, if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface** *interface interface-id*<br>**Example:**<br>`Device(config)# interface fortygigabitEthernet 1/0/10` | Enters interface configuration mode and selects the link to configure. |
| **Step 4** | **link debounce** [**time** *debounce-time*]<br>**Example:**<br>`Device(config-if)#link debounce time 360` | Configures the debounce timer in milli seconds. Maximum allowed timer is 1200 ms. The default port debounce timer is 240 ms<br>Link Debounce Timer works in multiples of 120 ms. If the configured timer value is above multiple of 120ms, the system uses the last multiple value of 120ms. |
| **Step 5** | **end**<br>**Example:**<br>`Device(config-if)#  end` | Returns to privileged EXEC mode. |
| **Step 6** | **show interface** *interface-id* **debounce** or **show interface debounce**<br>**Example:**<br>`Device#show interface debounce` | Verifies the configuration. |

# Example: Configuring the Link Debounce Timer

The following example shows how to enable Link Debounce Timer on a FortygigabitEthernet port.

```
Device(config)#inter fortygigabitEthernet 1/0/10
Device(config-if)#link debounce time 360
Warning: Enabling debounce feature causes link down detection to be delayed
```

The following examples show how to verify the Link Debounce Timer configuration.

```
Device#show inter fortygigabitEthernet 1/0/10 debounce

Port          Debounce time   Value(ms)
Fo1/0/10      enable               360


Device#show interfaces debounce

Port          Debounce time   Value(ms)
Fo1/0/1       disable
Fo1/0/2       disable
Fo1/0/3       enable          1200
Fo1/0/4       disable
Fo1/0/5       disable
Fo1/0/6       disable
Fo1/0/7       disable
Fo1/0/8       disable
Fo1/0/9       disable
Fo1/0/10      enable          360
Fo1/0/11      disable
Fo1/0/12      enable          240
Fo1/0/13      disable
```

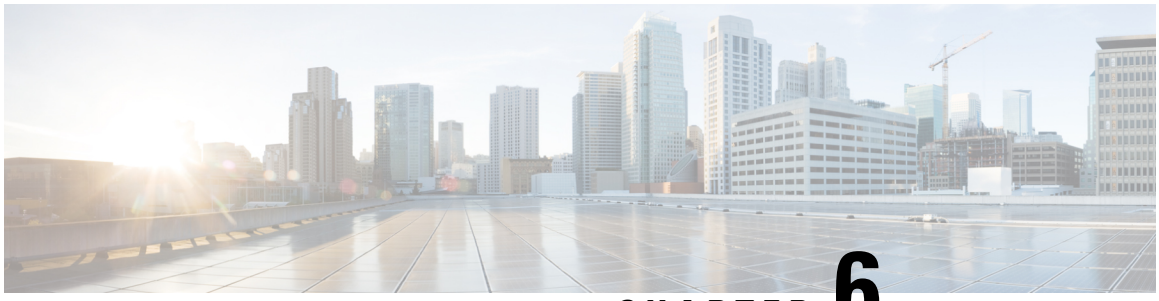
# Feature History for Link Debounce Timer

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 Amsterdam 17.3.3 | Link Debounce Timer | The Link Debounce Timer feature delays notification of a link down status change. Delayed notification of a linkdown status change can decrease traffic loss due to network reconfiguration when network ethernet portexperiences minor faults in the link. |

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/.



CHAPTER 6

# Configuring System MTU



## Restrictions for System MTU

On Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2), the following restrictions are applicable:

- If no protocol-specific MTU configuration is present, Per-Port MTU is used as protocol-specific MTU. In case Per-Port MTU is not configured, System MTU is used as protocol-specific MTU.
- Ingress and egress Layer 2 MTU is derived from Per-Port MTU. If Per-Port MTU is not configured, System MTU is used
- On ingress ports configured with Layer 2 MTU, if packets exceed the configured MTU size, then the packets are dropped.
- Layer 2 MTU configurations are not enforced for egress frames.

## 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. The maximum value of System MTU for Catalyst is 9216 bytes.

### System MTU Value Application

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

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

# How to Configure MTU

## Configuring the System MTU

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

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **system mtu** *bytes*<br>**Example:**<br>`Device(config)# system mtu 1900` | (Optional) Changes the MTU size for all interfaces. |
| **Step 4** | **end**<br>**Example:**<br>`Device(config)# end` | Returns to privileged EXEC mode. |
| **Step 5** | **copy running-config startup-config**<br>**Example:**<br>`Device# copy running-config startup-config` | Saves your entries in the configuration file. |
| **Step 6** | **show system mtu**<br>**Example:**<br>`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**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 2** | **interface** *interface*<br>**Example:**<br>`Device(config)# interface gigabitethernet0/0` | Enters interface configuration mode. |
| **Step 3** | **ip mtu** *bytes*<br>**Example:**<br>`Device(config-if)# ip mtu 68` | Changes the IPv4 MTU size |
| **Step 4** | **ipv6 mtu** *bytes*<br>**Example:**<br>`Device(config-if)# ipv6 mtu 1280` | (Optional) Changes the IPv6 MTU size. |
| **Step 5** | **end**<br>**Example:**<br>`Device(config-if)# end` | Returns to privileged EXEC mode. |
| **Step 6** | **copy running-config startup-config**<br>**Example:**<br>`Device# copy running-config startup-config` | Saves your entries in the configuration file. |
| **Step 7** | **show system mtu**<br>**Example:**<br>`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 9600 Series Switches)* |

### Standards and RFCs

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

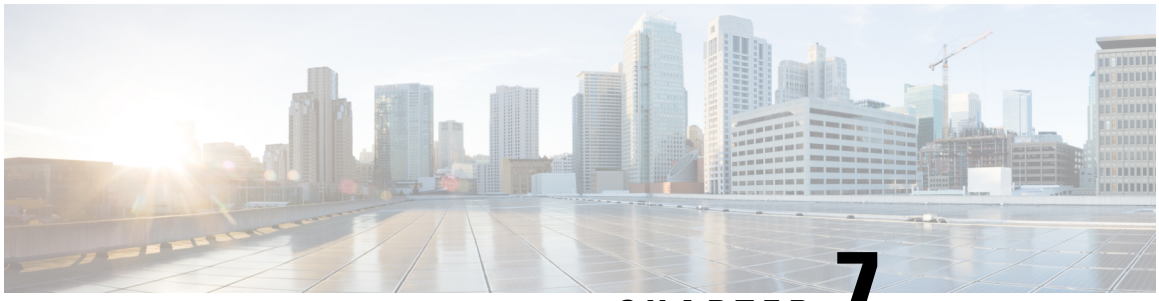
# Feature History for System MTU

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 Gibraltar 16.11.1 | System MTU | System MTU defines the maximum transmission unit size for frames transmitted on all interfaces of a switch. |

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.

Per-port MTU can be configured in the range of 1500-9216 bytes.

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

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

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

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

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

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

# Configuring Per-Port MTU

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

**Procedure**

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

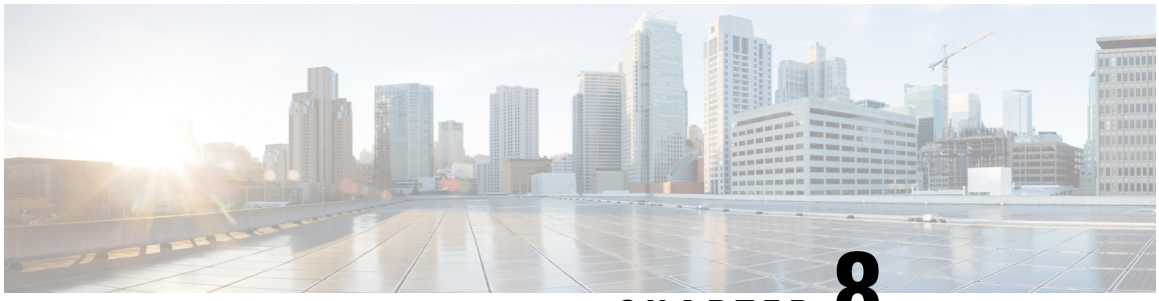
# Feature History for Per-Port MTU

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.<br>. |

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 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 7: 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. |

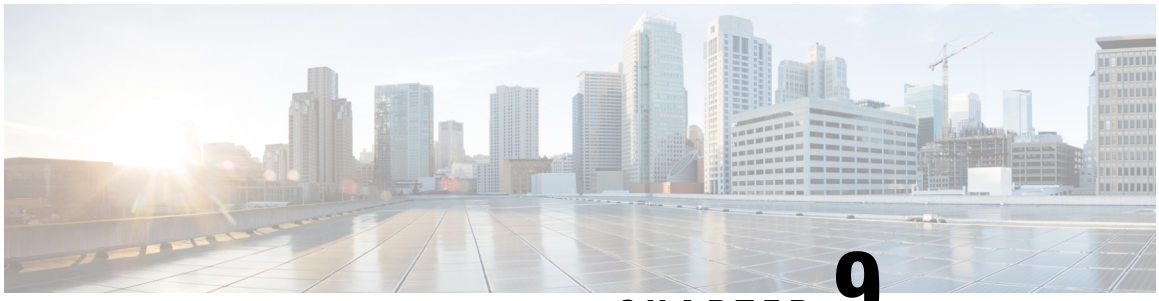
# Feature History for Configuring an External Bluetooth Dongle

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 Amsterdam 17.1.1 | 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. |

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

# M2 SATA Module



## M2 SATA Module on Cisco Catalyst 9600 Series Supervisor

Cisco Catalyst 9600 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 9600 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 use M2 SATA to boot images from ROMMON.
- You cannot upgrade the firmware on the M2 SATA drive.
- The M2 SATA device is not accessible from ROMMON. Hence you cannot perform any operations on the SATA device from ROMMON mode.

# 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 . 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. |

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)
```

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
NAME: "Chassis", DESCR: "Cisco Catalyst 9600 Series 6 Slot Chassis"
PID: C9606R            , VID: V00  , SN: FXS2231Q32N

NAME: "Slot 2 Linecard", DESCR: "48-Port 10GE / 25GE"
PID: C9600-LC-48YL     , VID: V00  , SN: CAT2232L0NJ

NAME: "TwentyFiveGigE2/0/1", DESCR: "10GE CU5M"
PID: QSFP-4SFP10G-CU5M   , VID: V03  , SN: MDM17350075-CH3

NAME: "TwentyFiveGigE2/0/2", DESCR: "10GE CU1M"
PID: SFP-H10GB-CU1M      , VID: V03  , SN: TED2143A0VQ

NAME: "TwentyFiveGigE2/0/3", DESCR: "10GE CU1M"
PID: SFP-H10GB-CU1M      , VID: V03  , SN: TED2143A0VQ

NAME: "TwentyFiveGigE2/0/4", DESCR: "10GE CU1M"
PID: SFP-H10GB-CU1M      , VID: V03  , SN: TED2143A0LU

NAME: "TwentyFiveGigE2/0/5", DESCR: "10GE CU1M"
PID: SFP-H10GB-CU1M      , VID: V03  , SN: TED2143A0LU

<output truncated>
```

# Feature History and Information for M2 SATA Module

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

| Feature Name | Release | Feature Information |
|---|---|---|
| M2 SATA Module | Cisco IOS XE Gibraltar 16.11.1 | 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. |