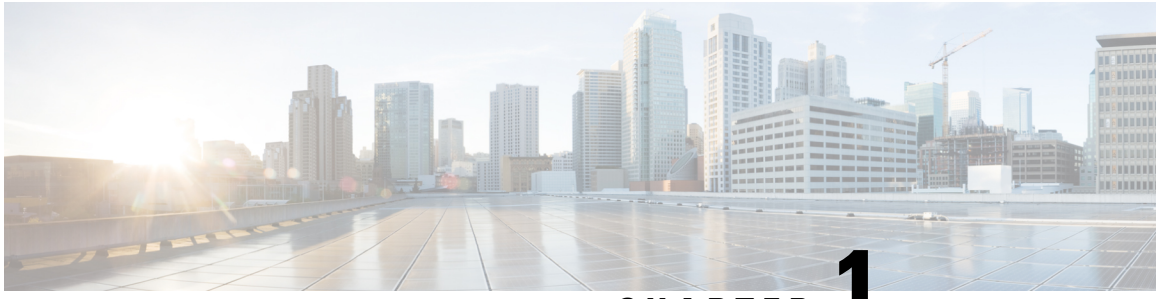
Interface and Hardware Configuration Guide, Cisco IOS Release 15.2(7)Ex (Catalyst 1000 Switches)

First Published: 2019-12-25

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

Restrictions for Configuring Interface Characteristics

- Flex Links are not supported.
- Multi-chassis EtherChannel (MEC) is not supported.
- Due to hardware restrictions, no Layer 3 routed counters will be displayed for Layer 3 interfaces (SVI, Routed, and/or Layer 3 port channel).

Information About Configuring Interface Characteristics

Interface Types

This section describes the different types of interfaces supported by the switch. 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 switch running configuration. With VTP version 3, you can create extended-range VLANs in client or server mode. These VLANs are saved in the VLAN database.

Add ports to a VLAN by using the **switchport** interface configuration commands:

- 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.

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.
- VLAN membership of dynamic access ports is learned through incoming packets. By default, a dynamic access port is not a member of any VLAN, and forwarding to and from the port is enabled only when the VLAN membership of the port is discovered.

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 switch supports only IEEE 802.1Q trunk ports. 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.

Routed Ports

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

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

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

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 or bridging 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 switch. By default, an SVI is created for the default VLAN (VLAN 1) to permit remote switch 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 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.

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 switches or between switches 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 or multiple access ports into one logical access 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, 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 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.

Dual-Purpose Uplink Ports

Note Dual-purpose uplink ports are supported on C1000-8T, C1000-8P, C1000-8FP, C1000FE-24T, C1000FE-24P, C1000FE-48T and C1000FE-48P models of the Cisco Catalyst 1000 Series Switches

Catalyst switches support dual-purpose uplink ports. Each uplink port is considered as a single interface with dual front ends—an RJ-45 connector and an small form-factor pluggable (SFP) module connector. The dual front ends are not redundant interfaces, and the switch activates only one connector of the pair.

By default, the switch dynamically selects the interface type that first links up. However, you can use the **media-type** interface configuration command to manually select the RJ-45 connector or the SFP module connector. For information about configuring speed and duplex settings for a dual-purpose uplink, see the Setting the Interface Speed and Duplex Parameters, on page 14 section.

Each uplink port has one LED which is located below the SFP module connector. The port LED is on for whichever uplink port is active. For more information about the LEDs, see the hardware installation guide.

Power over Ethernet Ports

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

- a Cisco pre-standard powered device (such as a Cisco IP Phone, a Cisco Aironet Access Point, or a Cisco Catalyst Access Point)
- an IEEE 802.3af and IEEE 802.3at-compliant powered device

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

Using the Switch USB Ports

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

USB Mini-Type B Console Port

The switch has the following console ports:

- USB mini-Type B console connection

  Note This is not supported on Cisco Catalyst 1000 Fast Ethernet Series Switches.

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

USB Type A Ports

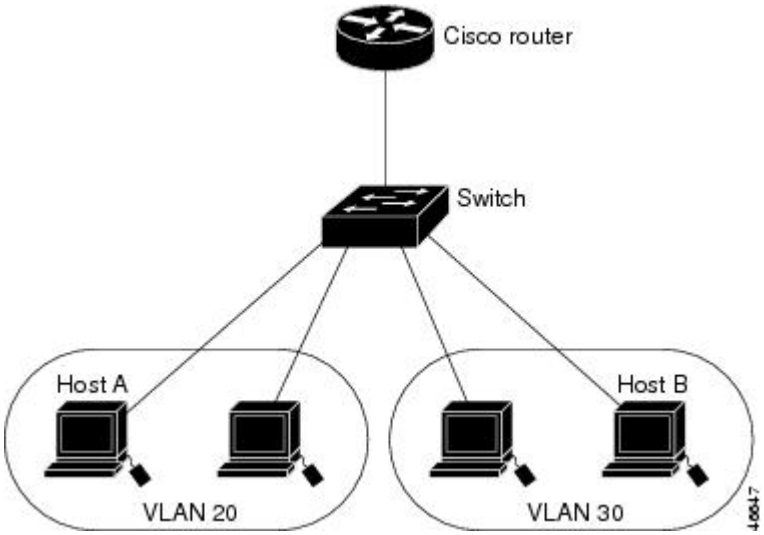
The USB Type A ports provide access to external USB flash devices, also known as thumb drives or USB keys. The switch supports Cisco 64 MB, 256 MB, 512 MB, 1 GB, 4 GB, and 8 GB flash drives. You can use standard Cisco IOS command- line interface (CLI) commands to read, write, erase, and copy to or from the flash device. You can also configure the switch to boot from the USB flash drive.

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.

In the following configuration example, when Host A in VLAN 20 sends data to Host B in VLAN 30, the data must go from Host A to the switch, to the router, back to the switch, and then to Host B.

Figure 1: Connecting VLANs with the Switch



With a standard Layer 2 switch, ports in different VLANs have to exchange information through a router.

Interface Configuration Mode

The switch supports these interface types:

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

You can also configure a range of interfaces.

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

- Type: Gigabit Ethernet (gigabitethernet or gi) for 10/100/1000 Mb/s Ethernet ports, or Fast Ethernet (fastethernet or fa) for 10/100 Mb/s Ethernet ports, or small form-factor pluggable (SFP) module Gigabit Ethernet interfaces (gigabitethernet or gi).
- Module number: The module or slot number on the switch (always 0).
- Port number: The interface number on the switch. The 10/100/1000 port numbers always begin at 1, starting with the far left port when facing the front of the switch, for example, gigabitethernet1/0/1 or gigabitethernet1/0/8. The 10/100 port numbers always begin at 1, starting with the far left port when facing the front of the switch, for example, fastethernet1/0/1 or fastethernet1/0/8. For a switch with 10/100/1000 ports and SFP module ports, SFP module ports are numbered consecutively following the 10/100/1000 ports.

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

Default Ethernet Interface Configuration

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

Table 1: Default Layer 2 Ethernet Interface Configuration

Feature	Default Setting
Operating mode	Layer 2 or switching mode (**switchport** command).
Allowed VLAN range	VLANs 1– 4094.
Default VLAN (for access ports)	VLAN 1.
Native VLAN (for IEEE 802.1Q trunks)	VLAN 1.
802.1p priority-tagged traffic	Drop all packets tagged with VLAN 0.
VLAN trunking	Switchport mode dynamic auto (supports DTP).
Port enable state	All ports are enabled.
Port description	None defined.
Speed	Autonegotiate. (Not supported on the 10-Gigabit interfaces.)
Duplex mode	Autonegotiate. (Not supported on the 10-Gigabit interfaces.)
Flow control	Flow control is set to **receive: off**. It is always off for sent packets.
EtherChannel (PAgP)	Disabled on all Ethernet ports.
Port blocking (unknown multicast and unknown unicast traffic)	Disabled (not blocked).
Broadcast, multicast, and unicast storm control	Disabled.
Protected port	Disabled.
Port security	Disabled.
Port Fast	Disabled.
Auto-MDIX	Enabled. **Note** The switch might not support a pre-standard powered device—such as Cisco IP phones and access points that do not fully support IEEE 802.3af—if that powered device is connected to the switch through a crossover cable. This is regardless of whether auto-MIDX is enabled on the switch port.

Feature	Default Setting
Power over Ethernet (PoE)	Enabled (auto).
Keepalive messages	Disabled on SFP module ports; enabled on all other ports.

Interface Speed and Duplex Mode

Ethernet interfaces on the switch operate at 10, 100, or 1000 Mb/s and in either full- or half-duplex mode. In full-duplex mode, two stations can send and receive traffic at the same time. Normally, 10-Mb/s ports operate in half-duplex mode, which means that stations can either receive or send traffic.

Switch modules include Gigabit Ethernet (10/100/1000-Mb/s) ports, Fast Ethernet (10/100-Mb/s) ports, and small form-factor pluggable (SFP) module slots supporting SFP modules.

Speed and Duplex Configuration Guidelines

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

- Do not disable Auto-Negotiation on PoE switches.
- Gigabit Ethernet (10/100/1000-Mb/s) and Fast Ethernet (10/100-Mb/s) ports support all speed options and all duplex options (auto, half, and full). However, Gigabit Ethernet ports operating at 1000 Mb/s do not support half-duplex mode.
- For SFP module ports, the speed and duplex CLI options change depending on the SFP module type.
- 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 switch can take up to 30 seconds to check for loops. The port LED is amber while STP reconfigures.
- As best practice, we suggest configuring the speed and duplex options on a link to auto or to fixed on both the ends. If one side of the link is configured to auto and the other side is configured to fixed, the link will not be up and this is expected.

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

IEEE 802.3x Flow Control

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

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

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.

How to Configure Interface Characteristics

Configuring Interfaces

These general instructions apply to all interface configuration processes.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** **Example:**	Identifies the interface type and the number of the connector.

	Command or Action	Purpose

	`Device(config)# interface gigabitethernet 1/0/1`
	Or
	`Device(config)# interface fastethernet 1/0/1`
		Note You do not need to add a space between the interface type and the interface number. For example, in the preceding line, you can specify either **gigabitethernet 1/0/1**, **gigabitethernet1/0/1**, **gi 1/0/1**, or **gi1/0/1**, or, **fastethernet 1/0/1**, **fastethernet1/0/1**, **fa 1/0/1**, or **fa1/0/1**.

Step 4	Follow each **interface** command with the interface configuration commands that the interface requires.	Defines the protocols and applications that will run on the interface. The commands are collected and applied to the interface when you enter another interface command or enter **end** to return to privileged EXEC mode.
Step 5	**interface range** or **interface range macro**	(Optional) Configures a range of interfaces. **Note** Interfaces configured in a range must be the same type and must be configured with the same feature options.
Step 6	**show interfaces**	Displays a list of all interfaces on or configured for the switch. A report is provided for each interface that the device supports or for the specified interface.

Adding a Description for an Interface

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 1/0/2`	Specifies the interface for which you are adding a description, and enter interface configuration mode.

	Command or Action	Purpose
	Or `Device(config)# interface fastethernet 1/0/2`	
Step 4	**description** *string* **Example:** `Device(config-if)# description Connects to Marketing`	Adds a description (up to 240 characters) for an interface.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show interfaces** *interface-id* **description**	Verifies your entry.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring a Range of Interfaces

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface range** {*port-range* | **macro** *macro_name*} **Example:** `Device(config)# interface range macro`	Specifies the range of interfaces (VLANs or physical ports) to be configured, and enter interface-range configuration mode. • You can use the **interface range** command to configure up to five port ranges or a previously defined macro.

	Command or Action	Purpose
		• The **macro** variable is explained in the section on *Configuring and Using Interface Range Macros.* • In a comma-separated *port-range*, you must enter the interface type for each entry and enter spaces before and after the comma. • In a hyphen-separated *port-range*, you do not need to re-enter the interface type, but you must enter a space before the hyphen. **Note** Use the regular configuration commands to apply the configuration parameters to all interfaces in the range. Each command is executed as it is entered.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**show interfaces** [*interface-id*] **Example:** `Device# show interfaces`	Verifies the configuration of the interfaces in the range.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring and Using Interface Range Macros

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**define interface-range** *macro_name interface-range*
	Example:
	`Device(config)# define interface-range enet_list gigabitethernet 1/0/1 - 2`
	Or
	`Device(config)# define interface-range enet_list fastethernet 1/0/1 - 2`
		Defines the interface-range macro, and save it in NVRAM.
		• The *macro_name* is a 32-character maximum character string.
		• A macro can contain up to five comma-separated interface ranges.
		• Each *interface-range* must consist of the same port type.
		Note Before you can use the **macro** keyword in the **interface range macro** global configuration command string, you must use the **define interface-range** global configuration command to define the macro.

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

Configuring Ethernet Interfaces

Setting the Interface Speed and Duplex Parameters

Procedure

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

Step 3	**interface** *interface-id*
	Example:
	`Device(config)# interface gigabitethernet 1/0/3`
	Or
	`Device(config)# interface fastethernet 1/0/3`
		Specifies the interface for which you are adding a description, and enter interface configuration mode.

Step 4	**speed** {**10** | **100** | **1000**} **Example:** `Device(config-if)# speed 10`	Enter the appropriate speed parameter for the interface: • Enter **10**, **100**, **1000** to set a specific speed for the interface.
Step 5	**duplex** {**auto** | **full** | **half**} **Example:** `Device(config-if)# duplex half`	This command is not available on a 10-Gigabit Ethernet interface. Enter the duplex parameter for the interface. Enable half-duplex mode (for interfaces operating only at 10 or 100 Mb/s). You cannot configure half-duplex mode for interfaces operating at 1000 Mb/s. You can configure the duplex setting when the speed is set to **auto**.
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

	Command or Action	Purpose

Step 7	**show interfaces** *interface-id*
	Example:
	`Device# show interfaces gigabitethernet 1/0/3`
	Or
	`Device# show interfaces fastethernet 1/0/3`
		Displays the interface speed and duplex mode configuration.

Step 8	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring IEEE 802.3x Flow Control

Procedure

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

Step 3	**interface** *interface-id*
	Example:
	`Device(config)# interface gigabitethernet 1/0/1`
	Or
	`Device(config)# interface fastethernet 1/0/1`
		Specifies the interface for which you are adding a description, and enter interface configuration mode.

Step 4	**flowcontrol** {**receive**} {**on** | **off** | **desired**} **Example:** `Device(config-if)# flowcontrol receive on`	Configures the flow control mode for the port.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Step 6	**show interfaces** *interface-id*
	Example:
	`Device# show interfaces gigabitethernet 1/0/1`
	Or
	`Device# show interfaces fastethernet 1/0/1`
		Verifies the interface flow control settings.

Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Shutting Down and Restarting the Interface

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** {**vlan** *vlan-id*} | {**gigabitethernet/fastethernet** *interface-id*} | {**port-channel** *port-channel-number*} **Example:** `Device(config)# interface gigabitethernet 1/0/2`	Selects the interface to be configured.

	Command or Action	Purpose
	Or `Device(config)# interface fastethernet 1/0/2`	
Step 4	**shutdown** **Example:** `Device(config-if)# shutdown`	Shuts down an interface.
Step 5	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Restarts an interface.
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 7	**show running-config** **Example:** `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.

Note This procedure is not applicable to Cisco Catalyst 1000 Fast Ethernet Series Switches.

Procedure

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

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

Configuring the 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.

Note This procedure is not applicable to Cisco Catalyst 1000 Fast Ethernet Series Switches.

Procedure

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

	Command or Action	Purpose
Step 3	**line console 0** **Example:** `Device(config)# line console 0`	Configures the console and enters line configuration mode.
Step 4	**usb-inactivity-timeout** *timeout-minutes* **Example:** `Device(config-line)# usb-inactivity-timeout 30`	Specify an inactivity timeout for the console port. The range is 1 to 240 minutes. The default is to have no timeout configured.
Step 5	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

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-number* **downshift module***module-number*	Displays the downshift status details of the specified interfaces and modules.
show interfaces *interface-id* **status** [**err-disabled**]	Displays interface status or a list of interfaces in the error-disabled state.
show interfaces [*interface-id*] **switchport**	Displays administrative and operational status of switching (nonrouting) ports. You can use this command to find out if a port is in routing or in switching mode.
show interfaces [*interface-id*] **description**	Displays the description configured on an interface or all interfaces and the interface status.
show ip interface [*interface-id*]	Displays the usability status of all interfaces configured for IP routing or the specified interface.
show interface [*interface-id*] **stats**	Displays the input and output packets by the switching path for the interface.

Command	Purpose
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.
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

Configuring a Range of Interfaces: Examples

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

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

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.

Configuring and Using Interface Range Macros: Examples

This example shows how to define an interface-range named *enet_list* to include ports 1 and 2 on switch 1 and to verify the macro configuration:

```
Device# configure terminal
Device(config)# define interface-range enet_list gigabitethernet 1/0/1 - 2
Device(config)# end
Device# show running-config | include define
define interface-range enet_list gigabitethernet 1/0/1 - 2
```

This 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)#
```

This 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#
```

Setting Interface Speed and Duplex Mode: Example

This example shows how to set the interface speed to 100 Mb/s and the duplex mode to half on a 10/100/1000 Mb/s port:

```
Device# configure terminal
Device(config)# interface gigabitethernet 1/0/3
Device(config-if)# speed 10
Device(config-if)# duplex half
```

This example shows how to set the interface speed to 100 Mb/s on a 10/100/1000 Mb/s port:

```
Device# configure terminal
Device(config)# interface gigabitethernet 1/0/2
Device(config-if)# speed 100
```

Configuring the Console Media Type: Example

This example disables the USB console media type and enables the RJ-45 console media type.

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

This example reverses the previous configuration and immediately activates any USB console that is connected.

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

Configuring the USB Inactivity Timeout: Example

This example configures the inactivity timeout to 30 minutes:

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

To disable the configuration, use these commands:

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

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 the switch has been disconnected and reconnected, a log similar to this appears:

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

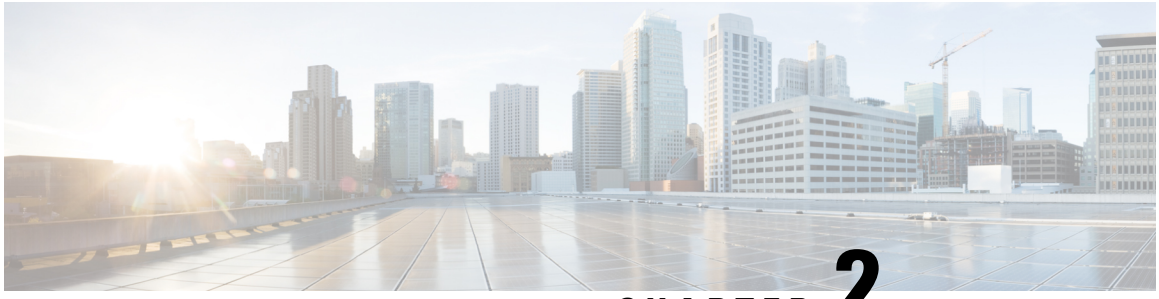
Feature History and Information for Configuring Interface Characteristics

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.

Use the Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 4: Feature History and Information for Configuring VLAN

Feature	Release	Feature Information
Configuring Interface Characteristics	Cisco IOS Release 15.2(7)E1	This feature was introduced.



CHAPTER 2

Configuring Auto-MDIX

Prerequisites for Auto-MDIX

If the interface is in Layer 3 mode and you want to want to configure Layer 2 parameters, you must first change the interface to Layer 2 mode. Enter the **switchport** interface configuration command without any parameters, to change the interface to Layer 2 mode. This shuts down the interface and then re-enables it, which may generate messages on the device to which the interface is connected. When you change an interface from Layer 3 mode to Layer 2 mode, previous configuration information related to the affected interface might be lost, and the interface is returned to its default configuration.

Restrictions for Auto-MDIX

- Automatic medium-dependent interface crossover (auto-MDIX) is supported on all 10/100/1000-Mb/s and on 10/100/1000BASE-TX small form-factor pluggable (SFP) module interfaces. It is not supported on 1000BASE-SX or -LX SFP module interfaces.
- The switch might not support a pre-standard powered device such as a Cisco IP phone or an access point that does not fully support IEEE 802.3af, if that powered device is connected to the switch through a crossover cable. This is regardless of whether auto-MIDX is enabled on the switch port or not.

Information About Auto-MDIX

Auto-MDIX on an Interface

When auto-MDIX is enabled on an interface, the interface automatically detects the required cable connection type (straight through or crossover) and configures the connection appropriately. When connecting devices without the auto-MDIX feature, you must use straight-through cables to connect to devices such as servers,

workstations, or routers and crossover cables to connect to other devices or repeaters. With auto-MDIX enabled, you can use either type of cable to connect to other devices, and the interface automatically corrects for any incorrect cabling. For more information about cabling requirements, see the hardware installation guide.

This feature is enabled by default.

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

Table 5: Link Conditions and Auto-MDIX Settings

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

How to Configure Auto-MDIX

Configuring Auto-MDIX on an Interface

To configure auto-MDIX on an interface, complete the following steps:

Procedure

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

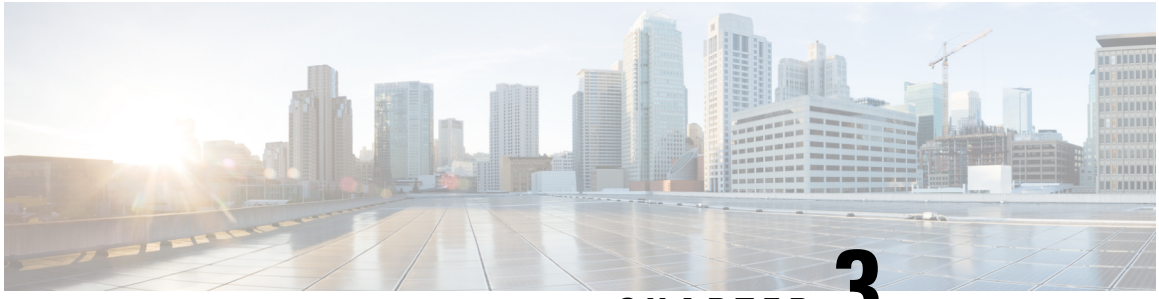
	Command or Action	Purpose
	Or `Device(config)# interface fastethernet 1/0/1`	
Step 4	**speed auto** **Example:** `Device(config-if)# speed auto`	Configures the interface to autonegotiate speed with the connected device.
Step 5	**duplex auto** **Example:** `Device(config-if)# duplex auto`	Configures the interface to autonegotiate duplex mode with the connected device.
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Feature History and Information for Auto-MDIX

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.

Use the Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Feature Name	Releases	Feature Information
Auto-MDIX	Cisco IOS Release 15.2(7)E1	This feature was introduced.



CHAPTER 3

Configuring LLDP, LLDP-MED, and Wired Location Service

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 and Cisco Medianet

When you configure LLDP or CDP location information on a per-port basis, remote devices can send Cisco Medianet location information to the device.

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 device 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; If no value is specified, the maximum is allowed (30 W).

- Inventory management TLV

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

- Location TLV

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

  - Civic location information

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

  - ELIN location information

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

Default LLDP Configuration

Table 6: Default LLDP Configuration

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

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**.

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

Enabling LLDP

Procedure

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

Step 4	**interface** *interface-id*
	Example:
	`Device(config)# interface gigabitethernet 1/0/1`
	Or
	`Device(config)# interface fastethernet 1/0/1`
		Specifies the interface on which you are enabling LLDP, and enter interface configuration mode.

Step 5	**lldp transmit** **Example:** `Device(config-if)# lldp transmit`	Enables the interface to send LLDP packets.

	Command or Action	Purpose
Step 6	**lldp receive** **Example:** `Device(config-if)# lldp receive`	Enables the interface to receive LLDP packets.
Step 7	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 8	**show lldp** **Example:** `Device# show lldp`	Verifies the configuration.
Step 9	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring LLDP Characteristics

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

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**lldp holdtime** *seconds* **Example:** `Device(config)# lldp holdtime 120`	(Optional) Specifies the amount of time a receiving device should hold the information from your device before discarding it. The range is 0 to 65535 seconds; the default is 120 seconds.

	Command or Action	Purpose
Step 4	**lldp reinit** *delay* **Example:** `Device(config)# lldp reinit 2`	(Optional) Specifies the delay time in seconds for LLDP to initialize on an interface. The range is 2 to 5 seconds; the default is 2 seconds.
Step 5	**lldp timer** *rate* **Example:** `Device(config)# lldp timer 30`	(Optional) Sets the sending frequency of LLDP updates in seconds. The range is 5 to 65534 seconds; the default is 30 seconds.
Step 6	**lldp tlv-select** **Example:** `Device(config)# tlv-select`	(Optional) Specifies the LLDP TLVs to send or receive.

Step 7	**interface** *interface-id*
	Example:
	`Device(config)# interface gigabitethernet 1/0/1`
	Or
	`Device(config)# interface fastethernet 1/0/1`
		Specifies the interface on which you are enabling LLDP, and enter interface configuration mode.

Step 8	**lldp med-tlv-select** **Example:** `Device(config-if)# lldp med-tlv-select inventory management`	(Optional) Specifies the LLDP-MED TLVs to send or receive.
Step 9	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 10	**show lldp** **Example:** `Device# show lldp`	Verifies the configuration.
Step 11	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring LLDP-MED TLVs

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

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

Table 7: LLDP-MED TLVs

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

Follow these steps to enable a TLV on an interface:

Procedure

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

Step 3	**interface** *interface-id*
	Example:
	`Device(config)# interface gigabitethernet 1/0/1`
	Or
	`Device(config)# interface fastethernet 1/0/1`
		Specifies the interface on which you are enabling LLDP, and enter interface configuration mode.

Step 4	**lldp med-tlv-select** **Example:** `Device(config-if)# lldp med-tlv-select inventory management`	Specifies the TLV to enable.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring Network-Policy TLV

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**network-policy profile** *profile number* **Example:** `Device(config)# network-policy profile 1`	Specifies the network-policy profile number, and enter network-policy configuration mode. The range is 1 to 4294967295.
Step 4	{**voice** | **voice-signaling**} **vlan** [*vlan-id* {**cos** *cvalue* | **dscp** *dvalue*}] | [[**dot1p** {**cos** *cvalue* | **dscp** *dvalue*}] | **none** | **untagged**] **Example:** `Device(config-network-policy)# voice vlan 100 cos 4`	Configures the policy attributes: • **voice**—Specifies the voice application type. • **voice-signaling**—Specifies the voice-signaling application type. • **vlan**—Specifies the native VLAN for voice traffic. • *vlan-id*—(Optional) Specifies the VLAN for voice traffic. The range is 1 to 4094. • **cos** *cvalue*—(Optional) Specifies the Layer 2 priority class of service (CoS) for the configured VLAN. The range is 0 to 7; the default is 5. • **dscp** *dvalue*—(Optional) Specifies the differentiated services code point (DSCP) value for the configured VLAN. The range is 0 to 63; the default is 46. • **dot1p**—(Optional) Configures the telephone to use IEEE 802.1p priority tagging and use VLAN 0 (the native VLAN). • **none**—(Optional) Do not instruct the IP telephone about the voice VLAN. The telephone uses the configuration from the telephone key pad.

	Command or Action	Purpose
		• **untagged**—(Optional) Configures the telephone to send untagged voice traffic. This is the default for the telephone.
Step 5	**exit** **Example:** `Device(config-network-policy)# exit`	Returns to global configuration mode.

Step 6	**interface** *interface-id*
	Example:
	`Device(config)# interface gigabitethernet 1/0/1`
	Or
	`Device(config)# interface fastethernet 1/0/1`
		Specifies the interface on which you are configuring a network-policy profile, and enter interface configuration mode.

Step 7	**network-policy** *profile number* **Example:** `Device(config-if)# network-policy 1`	Specifies the network-policy profile number.
Step 8	**lldp med-tlv-select network-policy** **Example:** `Device(config-if)# lldp med-tlv-select network-policy`	Specifies the network-policy TLV.
Step 9	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 10	**show network-policy profile** **Example:** `Device# show network-policy profile`	Verifies the configuration.
Step 11	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

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

Example: Configuring Network-Policy TLV

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

```
Device# configure terminal
Device(config)# network-policy 1
Device(config-network-policy)# voice vlan 100 cos 4
Device(config-network-policy)# exit
Device(config)# interface gigabitethernet 1/0/1
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

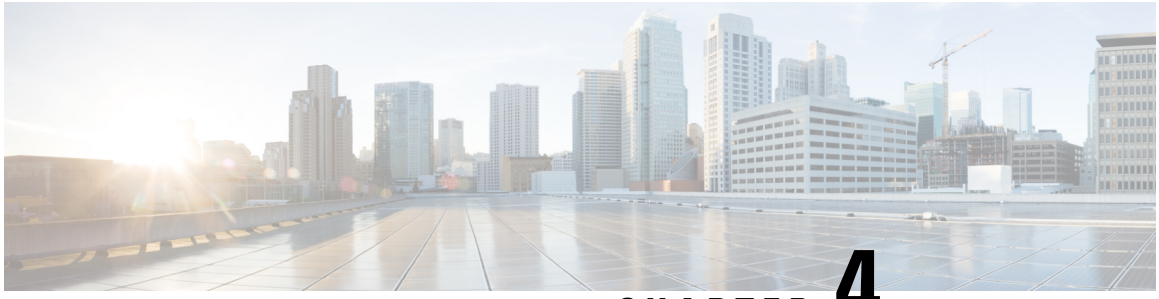
Commands for monitoring and maintaining LLDP, LLDP-MED, and wired location service.

Command	Description
clear lldp counters	Resets the traffic counters to zero.
clear lldp table	Deletes the LLDP neighbor information table.
clear nmsp statistics	Clears the NMSP statistic counters.
show lldp	Displays global information, such as frequency of transmissions, the holdtime for packets being sent, and the delay time before LLDP initializes on an interface.
show lldp entry *entry-name*	Displays information about a specific neighbor. You can enter an asterisk (*) to display all neighbors, or you can enter the neighbor name.

Command	Description
show lldp interface [*interface-id*]	Displays information about interfaces with LLDP enabled. You can limit the display to a specific interface.
show lldp neighbors [*interface-id*] [**detail**]	Displays information about neighbors, including device type, interface type and number, holdtime settings, capabilities, and port ID. You can limit the display to neighbors of a specific interface or expand the display for more detailed information.
show lldp traffic	Displays LLDP counters, including the number of packets sent and received, number of packets discarded, and number of unrecognized TLVs.
show location admin-tag *string*	Displays the location information for the specified administrative tag or site.
show location civic-location identifier *id*	Displays the location information for a specific global civic location.
show location elin-location identifier *id*	Displays the location information for an emergency location
show network-policy profile	Displays the configured network-policy profiles.
show nmsp	Displays the NMSP information

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

Release	Modification
Cisco IOS Release 15.2(7)E1	This feature was introduced.



CHAPTER 4

Configuring System MTU

Information About the MTU

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

You can change the MTU size to support switched jumbo frames on all Gigabit Ethernet and 10-Gigabit Ethernet interfaces and to support routed frames on all routed ports.

The default maximum transmission unit (MTU) size for frames received and transmitted on all interfaces is 1500 bytes. You can increase the MTU size to support jumbo frames on all Gigabit Ethernet interfaces by using the **system mtu jumbo** global configuration command.

Gigabit Ethernet ports are not affected by the **system mtu** command; 10/100 ports are not affected by the **system mtu jumbo** command. If you do not configure the **system mtu jumbo** command, the setting of the **system mtu** command applies to all Gigabit Ethernet interfaces.

How to Configure System MTU Sizes

Configuring the System MTU

Beginning in privileged EXEC mode, follow these steps to change the MTU size.

Procedure

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

	Command or Action	Purpose
Step 2	**system mtu** *bytes* **Example:** `Device(config)# system mtu 1500`	(Optional) Change the MTU size for all interfaces on the switch stack. Enter 1500, 2026 or jumbo to specify the MTU size. The MTU value of **jumbo** is 10218. **routing** sets the routing MTU for the system.
Step 3	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 4	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves your entries in the configuration file.
Step 5	**do show system mtu** **Example:** `Device# do show system mtu`	

Configuration Examples for System MTU

This example shows how to set the maximum packet size for a port to 1500 bytes:

```
Device(config)# system mtu 1500
```

This is an example of output from the **show system mtu** command:

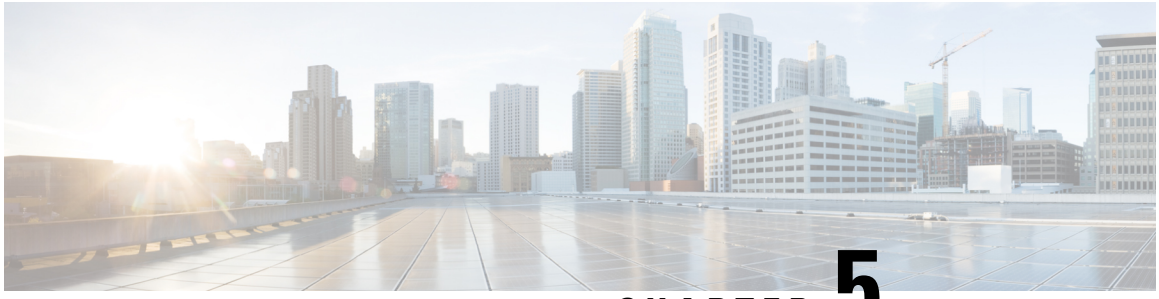
```
Device# show system mtu
System MTU size is 1500 bytes.
```

Feature Information for MTU

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.

Use the Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Feature Name	Releases	Feature Information
MTU	Cisco IOS Release 15.2(7)E1	This feature was introduced.



CHAPTER 5

Configuring Power over Ethernet

Information About PoE

Power over Ethernet Ports

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

- a Cisco pre-standard powered device (such as a Cisco IP Phone)
- an IEEE 802.3af and IEEE 802.3at-compliant powered device

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

Supported Protocols and Standards

The switch uses these protocols and standards to support PoE:

- CDP with power consumption: The powered switch notifies the device of the amount of power it is consuming. The device does not reply to the power-consumption messages. The device can only supply power to or remove power from the PoE port.
- Cisco intelligent power management: The powered device and the switch negotiate through power-negotiation CDP messages for an agreed-upon power-consumption level. The negotiation allows a high-power Cisco powered device, which consumes more than 7 W, to operate at its highest power mode. The powered device first boots up in low-power mode, consumes less than 7 W, and negotiates to obtain enough power to operate in high-power mode. The device changes to high-power mode only when it receives confirmation from the switch.

  High-power devices can operate in low-power mode on switches that do not support power-negotiation CDP.

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

- IEEE 802.3af and IEEE 802.3at: The major features of this standard are powered-device discovery, power administration, disconnect detection, and optional powered-device power classification. For more information, see the standard.

Powered-Device Detection and Initial Power Allocation

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

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

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

Table 8: IEEE Power Classifications

Class	Maximum Power Level Required from the Device
0 (class status unknown)	15.4 W
1	4 W
2	7 W
3	15.4 W
4	30 W (For IEEE 802.3at Type 2 powered devices)

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

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

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

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

Note The initial allocation for Class 0, Class 3, and Class 4 powered devices is 15.4 W. When a device starts up and uses CDP or LLDP to send a request for more than 15.4 W, it can be allocated up to the maximum of 30 W.

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

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

Power Management Modes

The device supports these PoE modes:

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

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

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

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

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

  You can specify the maximum wattage that is allowed on the port. If the IEEE class maximum wattage of the powered device is greater than the configured maximum value, the device does not provide power to the port. If you do not specify a wattage, the device delivers the maximum value. Use the **auto** setting on any PoE port. The auto mode is the default setting.

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

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

Note In interface mode, the power consumption of a device cannot exceed the power supplied to the static port.

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

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

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

Power Monitoring and Power Policing

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

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

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

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

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

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

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

   If error recovery is disabled, you can manually re-enable the PoE port by using the **shutdown** and **no shutdown** interface configuration commands.
4. If policing is disabled, no action occurs when the powered device consumes more than the maximum power allocation on the PoE port, which could adversely affect the device.

Maximum Power Allocation (Cutoff Power) on a PoE Port

When power policing is enabled, the device determines one of the these values as the cutoff power on the PoE port in this order:

1. Manually when you set the user-defined power level that limits the power allowed on the port by using the **power inline auto max** *max-wattage* or the **power inline static max** *max-wattage* interface configuration command
2. Automatically when the device sets the power usage of the device by using CDP power negotiation.

Use the first or second method in the previous list to manually configure the cutoff-power value by entering the **power inline consumption default** *wattage* or the **power inline** [**auto** | **static max**] *max-wattage* command.

If you do not manually configure the cutoff-power value, the device automatically determines it by using CDP power negotiation or the device IEEE classification and LLDP power negotiation. If CDP or LLDP are not enabled, the default value of 30 W is applied. However without CDP or LLDP, the device does not allow devices to consume more than 15.4 W of power because values from 15400 to 30000 mW are only allocated based on CDP or LLDP requests. If a powered device consumes more than 15.4 W without CDP or LLDP negotiation, the device might be in violation of the maximum current (*Imax*) limitation and might experience an *Icut* fault for drawing more current than the maximum. The port remains in the fault state for a time before attempting to power on again. If the port continuously draws more than 15.4 W, the cycle repeats.

Note When a powered device connected to a PoE+ port restarts and sends a CDP or LLDP packet with a power TLV, the device locks to the power-negotiation protocol of that first packet and does not respond to power requests from the other protocol. For example, if the device is locked to CDP, it does not provide power to devices that send LLDP requests. If CDP is disabled after the device has locked on it, the device does not respond to LLDP power requests and can no longer power on any accessories. In this case, you should restart the powered device.

Power Consumption Values

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

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

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

Note In interface mode, the power consumption of a device cannot exceed the power supplied to the static port.

For example, if you configure power supply to the port at 6000 mW (**power inline static6000** interface configuration command), do not configure power consumption by a device at 8000 mW on the same port (**power inline consumption8000** interface configuration command).

Persistent PoE

Persistent PoE provides uninterrupted power to connected devices even when the switch is booting.

How to Configure PoE

Configuring a Power Management Mode on a PoE Port

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

Procedure

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

Step 3	**interface** *interface-id*
	Example:
	`Device(config)# interface gigabitethernet 1/0/1`
	Or
	`Device(config)# interface fastethernet 1/0/1`
		Specifies the physical port to be configured, and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**power inline** {**auto** [**max** *max-wattage*] | **never** | **static** [**max** *max-wattage*]} **Example:** `Device(config-if)# power inline auto`	Configures the PoE mode on the port. The keywords have these meanings: • **auto**—Enables powered-device detection. If enough power is available, automatically allocates power to the PoE port after device detection. This is the default setting. • **max** *max-wattage*—Limits the power allowed on the port. The range is 4000 to 30000 mW. If no value is specified, the maximum is allowed. • **never** —Disables device detection, and disable power to the port. **Note** If a port has a Cisco powered device connected to it, do not use the **power inline never** command to configure the port. A false link-up can occur, placing the port into the error-disabled state. • **static**—Enables powered-device detection. Pre-allocate (reserve) power for a port before the device discovers the powered device. The device reserves power for this port even when no device is connected and guarantees that power will be provided upon device detection. **Note** Configure power values in multiples of 100. For example, you can configure 7400 mW, but do not configure 7386 mW or 7421 mW. The device allocates power to a port configured in static mode before it allocates power to a port configured in auto mode.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show power inline** [*interface-id* | **module** *switch-number*] **Example:** `Device# show power inline`	Displays PoE status for a device for the specified interface.

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

Configuring Persistent PoE

To configure persistent PoE, perform the following steps:

Note You will need to configure the **poe-ha** command before connecting the PD, or you will need to manually shut/unshut the port after configuring **poe-ha**.

If you want to reload the switch, ensure that the persistent PoE configuration is first saved. This is necessary to preserve the configuration.

Procedure

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

Step 3	**interface** *interface-id*
	Example:
	`Device(config)# interface gigabitethernet 1/0/1`
	Or
	`Device(config)# interface fastethernet 1/0/1`
		Specifies the physical port to be configured, and enters interface configuration mode.

Step 4	**power inline port poe-ha** **Example:** `Device(config-if)# power inline port poe-ha`	Configures persistent PoE.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Budgeting Power for Devices Connected to a PoE Port

When Cisco powered devices are connected to PoE ports, the device uses Cisco Discovery Protocol (CDP) or Link Layer Discovery Protocol (LLDP) to determine the *protocol-specific* power consumption of the devices, and the device adjusts the power budget accordingly. This does not apply to IEEE third-party powered devices. For these devices, when the device grants a power request, the device adjusts the power budget according to the powered-device IEEE classification. If the powered device is a class 0 (class status unknown) or a class 3, the device budgets 15,400 mW for the device, regardless of the CDP-specific amount of power needed. If the powered device reports a higher class than its CDP-specific consumption or does not support power classification (defaults to class 0), the device can power fewer devices because it uses the IEEE class information to track the global power budget.

By using the **power inline consumption** *wattage* interface configuration command, you can override the default power requirement specified by the IEEE classification. The difference between what is mandated by the IEEE classification and what is actually needed by the device is reclaimed into the global power budget for use by additional devices. You can then extend the device power budget and use it more effectively.

Caution You should carefully plan your device power budget, enable the power monitoring feature, and make certain not to oversubscribe the power supply.

Note When you manually configure the power budget, you must also consider the power loss over the cable between the device and the powered device.

Note In interface mode, the power consumption of a device cannot exceed the power supplied to the static port.

For example, if you configure power supply to the port at 6000 mW (**power inline static6000** interface configuration command), do not configure power consumption by a device at 8000 mW on the same port (**power inline consumption8000** interface configuration command).

Budgeting Power to All PoE ports

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**no cdp run** **Example:** `Device(config)# no cdp run`	(Optional) Disables CDP.
Step 4	**power inline consumption default** *wattage* **Example:** `Device(config)# power inline consumption default 5000`	Configures the power consumption of powered devices connected to each PoE port. The range for each device is 4000 to 15400 mW (PoE+). The default is 15400 mW.
Step 5	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 6	**show power inline consumption** **Example:** `Device# show power inline consumption`	Displays the power consumption status.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Budgeting Power to a Specific PoE Port

Procedure

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

	Command or Action	Purpose
Step 3	**no cdp run** **Example:** `Device(config)# no cdp run`	(Optional) Disables CDP.

Step 4	**interface** *interface-id*
	Example:
	`Device(config)# interface gigabitethernet 1/0/1`
	Or
	`Device(config)# interface fastethernet 1/0/1`
		Specifies the physical port to be configured, and enter interface configuration mode.

Step 5	**power inline consumption** *wattage* **Example:** `Device(config-if)# power inline consumption 5000`	Configures the power consumption of a powered device connected to a PoE port on the device. The range for each device is 4000 to 30000 mW (PoE+). The default is 15400 mW (PoE+).
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 7	**show power inline consumption** **Example:** `Device# show power inline consumption`	Displays the power consumption data.
Step 8	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring Power Policing

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

Note The power consumption is displayed in units of 0.5 W. For example, if a connected device draws 3.9 W, this feature will display 4.0 W power drawn.

Procedure

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

Step 3	**interface** *interface-id*
	Example:
	`Device(config)# interface gigabitethernet 1/0/1`
	Or
	`Device(config)# interface fastethernet 1/0/1`
		Specifies the physical port to be configured, and enter interface configuration mode.

Step 4	**power inline police** [**action** {**log** | **errdisable**}] **Example:** `Device(config-if)# power inline police`	If the real-time power consumption exceeds the maximum power allocation on the port, configures the device to take one of these actions: • **power inline police**—Shuts down the PoE port, turns off power to it, and puts it in the error-disabled state. **Note** You can enable error detection for the PoE error-disabled cause by using the **errdisable detect cause inline-power** global configuration command. You can also enable the timer to recover from the PoE error-disabled state by using the **errdisable recovery cause inline-power interval** *interval* global configuration command. • **power inline police action errdisable**—Turns off power to the port if the real-time power consumption exceeds the maximum power allocation on the port. • **power inline police action log**—Generates a syslog message while still providing power to the port.

	Command or Action	Purpose
		If you do not enter the **action log** keywords, the default action shuts down the port and puts the port in the error-disabled state.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.

Step 6	Use one of the following:
	• **errdisable detect cause inline-power**
	• **errdisable recovery cause inline-power**
	• **errdisable recovery interval** *interval*
	Example:
	`Device(config)# errdisable detect cause inline-power`
	`Device(config)# errdisable recovery cause inline-power`
	`Device(config)# errdisable recovery interval 100`
		(Optional) Enables error recovery from the PoE error-disabled state, and configures the PoE recover mechanism variables.
		By default, the recovery interval is 300 seconds.
		For **interval** *interval*, specifies the time in seconds to recover from the error-disabled state. The range is 30 to 86400.

Step 7	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.

Step 8	Use one of the following:
	• **show power inline police**
	• **show errdisable recovery**
	Example:
	`Device# show power inline police`
	`Device# show errdisable recovery`
		Displays the power monitoring status, and verify the error recovery settings.

Step 9	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Monitoring Power Status

Table 9: Show Commands for Power Status

Command	Purpose
show env power	(Optional) Displays the status of the internal power supplies for the switch.
show power inline [*interface-id*	Displays PoE status for an interface.
show power inline police	Displays the power policing data.

Note Use the**debug ilpower controller** privileged EXEC command to enable debugging of the platform-specific Power over Ethernet (PoE) software module on the switch in long message format. These messages include the Power Controller register reading. Use the **no** form of this command to disable debugging.

Configuration Examples for Configuring PoE

Budgeting Power: Example

When you enter one of the following commands, this caution message appears:

- [**no**] **power inline consumption default** *wattage* global configuration command
- [**no**] **power inline consumption** *wattage*

  interface configuration command

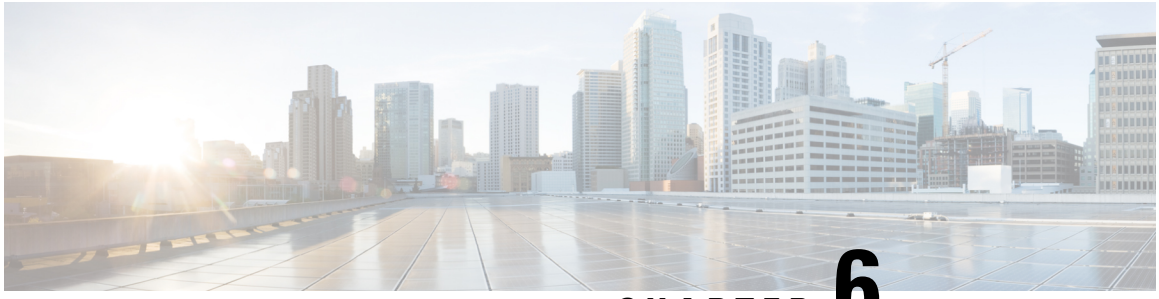
```
%CAUTION: Interface Gi0/1: Misconfiguring the 'power inline  consumption/allocation' command
 may cause damage to the
switch and void your warranty. Take precaution not to oversubscribe the power supply. It
is recommended to enable power
policing if the switch supports it. Refer to documentation.
```

Feature Information for PoE

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.

Use the Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Feature Name	Releases	Feature Information
PoE	Cisco IOS Release 15.2(7)E1	This feature was introduced.



CHAPTER 6

Configuring 2-event Classification

Information about 2-event Classification

When 2-event classification is configured and a class 4 device is detected, IOS allocates 30W without any CDP or LLDP negotiation. This means that even before the link comes up the class 4 power device gets 30W.

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

Once 2-event is enabled, the port resets automatically. Power budget allocation for a class-4 device will be 30W if 2-event classification is enabled on the port, else it will be 15.4W.

Configuring 2-event Classification

To configure the switch for a 2-event Classification, perform the steps given below:

Procedure

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

	Command or Action	Purpose
	`Device# configure terminal`	

Step 3	**interface** *interface-id*
	Example:
	`Device(config)# interface gigabitethernet 2/0/1`
	Or
	`Device(config)# interface fastethernet 1/0/1`
		Specifies the physical port to be configured, and enters interface configuration mode.

Step 4	**power inline port 2-event** **Example:** `Device(config-if)# power inline port 2-event`	Configures 2-event classification on the switch.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Example: Configuring 2-Event Classification

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

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

Additional References

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	Consolidated Platform Command Reference, Cisco IOS Release 15.2(7)Ex (Catalyst 1000 Switches)

MIBs

MIB	MIBs Link
All the supported MIBs for this release.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/support

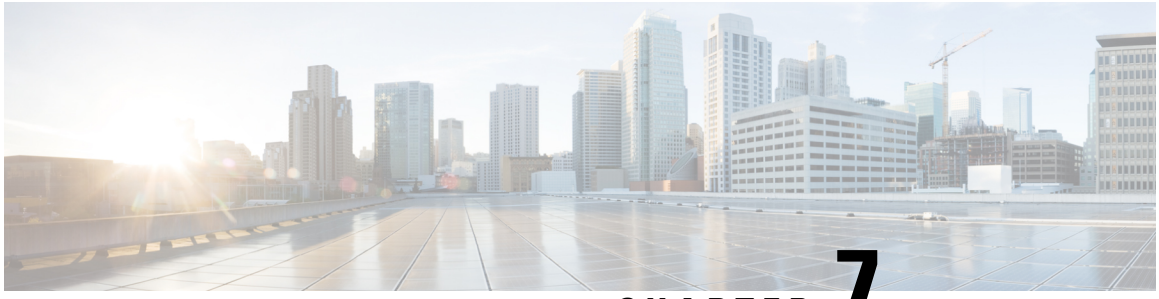
Feature History and Information for 2-event Classification

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.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 10: Feature Information for 2-event Classification

Feature Name	Releases	Feature Information
2-event Classification	Cisco IOS Release 15.2(7)E1	This feature was introduced.



CHAPTER 7

Configuring EEE

Prerequisites for EEE

Enable the Link Layer Discovery Protocol (LLDP) for devices that require longer wakeup times before they are able to accept data on their receive paths. Doing so enables the device to negotiate for extended system wakeup times from the transmitting link partner.

Restrictions for EEE

Changing the Energy Efficient Ethernet (EEE) configuration resets the interface because the device has to restart Layer 1 autonegotiation.

Information About EEE

EEE Overview

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

Default EEE Configuration

EEE is enabled by default.

How to Configure EEE

Enabling or Disabling EEE

You can enable or disable EEE on an interface that is connected to an EEE-capable link partner. Starting in the Privileged EXEC mode, complete the following steps to enable EEE:

Procedure

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

Step 2	**interface** *interface-id*
	Example:
	`Device(config)# interface gigabitethernet 1/0/1`
	Or
	`Device(config)# interface fastethernet 1/0/1`
		Specifies the interface to be configured, and enter interface configuration mode.

Step 3	**power efficient-ethernet auto** **Example:** `Device(config-if)# power efficient-ethernet auto`	Enables EEE on the specified interface. When EEE is enabled, the device advertises and autonegotiates EEE to its link partner. Enter the **no** form of the command to disable EEE.
Step 4	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 5	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Monitoring EEE

Table 11: Commands for Displaying EEE Settings

Command	Purpose
show eee capabilities interface *interface-id*	Displays EEE capabilities for the specified interface.
show eee status interface *interface-id*	Displays EEE status information for the specified interface.

The following are sample outputs of the **show eee** commands:

```
Device# show eee capabilities interface gigabitethernet 1/0/1
Gi0/1
EEE(efficient-ethernet): yes (100-Tx and 1000T auto)
Link Partner : yes (100-Tx and 1000T auto)

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

```
Device# show eee status interface gigabitethernet 1/0/1
Gi0/1 is up
EEE(efficient-ethernet): Operational
Rx LPI Status : Low Power
Tx LPI Status : Low Power
Wake Error Count : 0

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

Configuration Examples for EEE

The following example shows how to disable EEE on an interface:

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

Feature History and Information for EEE

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.

Use the Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Feature Name	Release	Feature Information
Energy Efficient Ethernet (EEE)	Cisco IOS Release 15.2(7)E1	This feature was introduced Energy Efficient Ethernet (EEE) is an IEEE 802.3az standard that is designed to reduce power consumption in Ethernet networks during idle periods.