Interface and Hardware Components Configuration Guide, Cisco IOS XE Bengaluru 17.5.x (Catalyst 9300 Switches)

First Published: 2021-04-01

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

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

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

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

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

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

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

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

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

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

© 2021 Cisco Systems, Inc. All rights reserved.

CONTENTS

CHAPTER 1 **Configuring Interface Characteristics** 1

- Information About Interface Characteristics 1
  - Interface Types 1
    - Port-Based VLANs 1
    - Switch Ports 2
  - Using the Switch USB Ports 6
    - USB Mini-Type B Console Port 6
    - Console Port Change Logs 6
    - USB Type A Port 7
    - Disabling USB Ports 7
  - Interface Connections 7
  - Interface Configuration Mode 8
  - Breakout Interfaces 9
    - Limitations for Breakout Interfaces 10
  - Default Ethernet Interface Configuration 10
  - Interface Speed and Duplex Mode 12
  - Speed and Duplex Configuration Guidelines 12
  - IEEE 802.3x Flow Control 13
  - Layer 3 Interfaces 13
- How to Configure Interface Characteristics 15
  - Configuring an Interface 15
  - Adding a Description for an Interface 16
  - Configuring a Range of Interfaces 17
  - Configuring and Using Interface Range Macros 18
  - Setting the Interface Speed and Duplex Parameters 19
  - Configuring a Breakout Interface 21

Configuring Forty Gigabit Ethernet Interface 23
Configuring the IEEE 802.3x Flow Control 23
Configuring a Layer 3 Interface 24
Configuring a Logical Layer 3 GRE Tunnel Interface 26
Configuring SVI Autostate Exclude 27
Shutting Down and Restarting an Interface 28
Configuring the Console Media Type 29
Configuring USB Inactivity Timeout 30
Disabling USB Ports 31
Monitoring Interface Characteristics 32
Monitoring Interface Status 32
Clearing and Resetting Interfaces and Counters 33
Configuration Examples for Interface Characteristics 33
Example: Adding a Description to an Interface 33
Example: Configuring Interfaces on a Stack-Capable Switch 34
Example: Configuring a Range of Interfaces 34
Example: Configuring and Using Interface Range Macros 34
Example: Setting Interface Speed and Duplex Mode 35
Example: Configuring a Layer 3 Interface 35
Example: Configuring a Breakout Interface 36
Example: Configuring the Console Media Type 38
Example: Configuring USB Inactivity Timeout 38
Additional References for Configuring Interface Characteristics 39
Feature History for Configuring Interface Characteristics 39

CHAPTER 2 **Configuring Auto-MDIX** 43
Prerequisites for Auto-MDIX 43
Restrictions for Auto-MDIX 43
Information About Configuring Auto-MDIX 44
Auto-MDIX on an Interface 44
How to Configure Auto-MDIX 44
Configuring Auto-MDIX on an Interface 44
Example for Configuring Auto-MDIX 45
Auto-MDIX and Operational State 46

Additional References for Auto-MDIX 46
Feature History for Auto-MDIX 46

CHAPTER 3 **Configuring Ethernet Management Port 49**
Prerequisites for Ethernet Management Port 49
Information About the Ethernet Management Port 49
Ethernet Management Port Direct Connection to a Device 49
Ethernet Management Port Connection to Stack Devices using a Hub 50
Ethernet Management Port and Routing 50
Supported Features on the Ethernet Management Port 51
How to Configure the Ethernet Management Port 52
Disabling and Enabling the Ethernet Management Port 52
Example for Configuring IP Address on Ethernet Management Interface 53
Additional References for Ethernet Management Port 53
Feature History for Ethernet Management Port 54

CHAPTER 4 **Checking Port Status and Connectivity 55**
Check Cable Status Using Time Domain Reflectometer 55
Running the TDR Test 56
TDR Guidelines 56
Feature History for Checking Port Status and Connectivity 57

CHAPTER 5 **Configuring LLDP, LLDP-MED, and Wired Location Service 59**
Restrictions for LLDP 59
Information About LLDP, LLDP-MED, and Wired Location Service 59
LLDP 59
LLDP Supported TLVs 60
LLDP-MED 60
LLDP-MED Supported TLVs 60
Wired Location Service 62
Default LLDP Configuration 63
How to Configure LLDP, LLDP-MED, and Wired Location Service 63
Enabling LLDP 63
Configuring LLDP Characteristics 64

Configuring LLDP-MED TLVs 66
Configuring Network-Policy TLV 68
Configuring Location TLV and Wired Location Service 70
Enabling Wired Location Service on the Device 72
Configuration Examples for LLDP, LLDP-MED, and Wired Location Service 73
Configuring Network-Policy TLV: Examples 73
Monitoring and Maintaining LLDP, LLDP-MED, and Wired Location Service 74
Additional References for LLDP, LLDP-MED, and Wired Location Service 75
Feature History for LLDP, LLDP-MED, and Wired Location Service 75

CHAPTER 6 **Configuring System MTU** 77
Restrictions for System MTU 77
Information About the MTU 77
System MTU Value Application 77
How to Configure MTU 78
Configuring the System MTU 78
Configuring Protocol-Specific MTU 79
Configuration Examples for System MTU 79
Example: Configuring Protocol-Specific MTU 79
Example: Configuring the System MTU 80
Additional References for System MTU 80
Feature History for System MTU 80

CHAPTER 7 **Configuring Per-Port MTU** 81
Restrictions for Per-Port MTU 81
Information About Per-Port MTU 81
Configuring Per-Port MTU 82
Example: Configuring Per-Port MTU 82
Example: Verifying Per-Port MTU 83
Example: Disabling Per-Port MTU 83
Feature History for Per-Port MTU 83

CHAPTER 8 **Configuring Internal Power Supplies** 85
Information About Internal Power Supplies 85

How to Configure Internal Power Supplies 85
Configuring Internal Power Supply 85
Monitoring Internal Power Supplies 86
Configuration Examples for Internal Power Supplies 86
Additional References for Internal Power Supplies 87
Feature History for Internal Power Supplies 87

CHAPTER 9 **Configuring the Cisco Expandable Power System 2200** 89
Restrictions for Configuring the Expandable Power System 2200 89
Information About Cisco Expandable Power System 2200 89
Cisco eXpandable Power System (XPS) 2200 Overview 89
XPS 2200 Power Supply Modes 90
RPS Mode 90
Stack Power Mode 91
Mixed Modes 92
XPS 2200 System Defaults 92
How to Configure the Cisco Expandable Power System 2200 93
Configuring System Names 93
Configuring XPS Ports 94
Configuring XPS Power Supplies 95
Monitoring and Maintaining the Cisco Expandable Power System 2200 96
Additional References for Cisco Expandable Power System 2200 96
Feature History for Cisco Expandable Power System 2200 97

CHAPTER 10 **Configuring EEE** 99
Restrictions for EEE 99
Information About EEE 99
EEE Overview 99
Default EEE Configuration 100
How to Configure EEE 100
Enabling or Disabling EEE 100
Monitoring EEE 101
Configuration Examples for Configuring EEE 102
Additional References for EEE 102

Feature History for Configuring EEE 102

CHAPTER 11 **Configuring Power over Ethernet 103**

Information About Power over Ethernet 103

PoE and PoE+ Ports 103

Supported Protocols and Standards 104

Powered-Device Detection and Initial Power Allocation 105

Power Management Modes 107

Cisco Universal Power Over Ethernet 109

How to Configure PoE and UPOE 109

Configuring a Power Management Mode on a PoE Port 110

Enabling Power on Signal and Spare Pairs 111

Configuring Power Policing 112

Enable the 802.3bt Mode on Type 3 UPOE Modules 114

Monitoring Power Status 114

Additional References for Power over Ethernet 118

Feature History for Power over Ethernet 118

CHAPTER 12 **Configuring Perpetual PoE and Fast POE 121**

Restrictions for Perpetual and Fast PoE 121

Perpetual POE 122

Fast POE 122

Configuring Perpetual and Fast POE 122

Example: Configuring Perpetual and Fast POE 123

Feature Information for Persistent and Fast PoE 124

CHAPTER 13 **Configuring 2-event Classification 125**

Restrictions for 2-event classification 125

Information about 2-event Classification 125

Configuring 2-event Classification 125

Example: Configuring 2-Event Classification 126

Feature History for 2-event Classification 126

CHAPTER 14 **Configuring Auto SmartPorts 129**

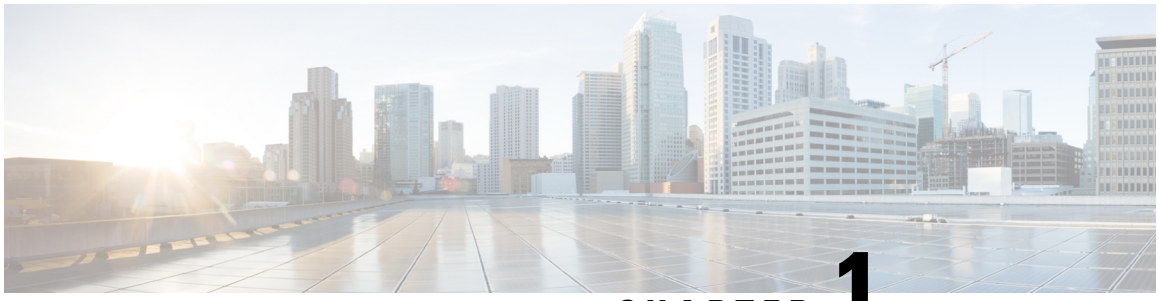
Restrictions for Auto SmartPorts 129
Information about Auto SmartPorts 129
Auto SmartPort Macros 130
Customizing Device Classifier 130
Commands executed by CISCO_LIGHT_AUTO_SMARTPORT 130
Enabling Auto SmartPort 131
Configuring Device Classifier Profiles 132
Configuring Mapping Between Event Triggers and Built-in Macros 134
Example: Enabling Auto SmartPorts 135
Example: Configuring Mapping Between Event Triggers and Built-in Macros 135
Example: Configuring Device Classifier Profiles 136
Feature Information for Auto SmartPorts 136

CHAPTER 15 **Configuring COAP Proxy Server** 139
Restrictions for the COAP Proxy Server 139
Information About the COAP Proxy Server 139
How to Configure the COAP Proxy Server 140
Configuring the COAP Proxy 140
Configuring COAP Endpoints 142
Configuration Examples for the COAP Proxy Server 143
Examples: Configuring the COAP Proxy Server 143
Monitoring COAP Proxy Server 147
Feature History for COAP 148

CHAPTER 16 **Configuring USB 3.0 SSD** 149
Information about USB 3.0 SSD 149
USB 3.0 SSD 149
File System on USB 3.0 SSD 150
Password Authentication on USB 3.0 SSD 150
How to Configure USB 3.0 SSD 150
Formatting USB 3.0 SSD 150
Unmounting USB 3.0 SSD from a Switch or a Switch Stack 150
Enabling Password Security on USB 3.0 SSD 151
Configuring USB 3.0 SSD Password on a Switch 152

Unlocking USB 3.0 SSD 153
Disabling Password Security on USB 3.0 SSD 153
Monitoring USB 3.0 SSD 153
Troubleshooting Tips 154
Troubleshooting USB 3.0 SSD Insertion and Removal 155
Troubleshooting Password Authentication 156
Configuration Examples for USB 3.0 SSD 157
Example: Displaying USB 3.0 SSD Authentication Status 157
Examples: Verifying the Filesystem 157
Examples: Verifying Physical Inventory Information 158
Examples: Verifying the Health of the Drive 158
Feature History for USB 3.0 SSD 159

CHAPTER 17 **Configuring an External USB Bluetooth Dongle** 161
Restrictions for Configuring an External USB Bluetooth Dongle 161
Information About External USB Bluetooth Dongle 161
Supported External USB Bluetooth Dongle 161
How to Configure an External USB Bluetooth Dongle on a Switch 162
Verifying Bluetooth Settings on a Switch 163
Feature History for Configuring an External Bluetooth Dongle 163



CHAPTER 1

Configuring Interface Characteristics

- Information About Interface Characteristics, on page 1
- How to Configure Interface Characteristics, on page 15
- Configuration Examples for Interface Characteristics, on page 33
- Additional References for Configuring Interface Characteristics, on page 39
- Feature History for Configuring Interface Characteristics, on page 39

Information About Interface Characteristics

The following sections provide information about interface characteristics.

Interface Types

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

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

Port-Based VLANs

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

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

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

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

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

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

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

Switch Ports

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

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

Access Ports

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

The types of access ports supported are:

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

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

Trunk Ports

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

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

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

Tunnel Ports

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

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

Routed Ports

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

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

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

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

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

Switch Virtual Interfaces

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

Note You cannot delete interface VLAN 1.

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

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

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

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

EtherChannel Port Groups

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

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

Network Modules

The device supports four network modules that include one Gigabit Ethernet, 10-Gigabit Ethernet, 25-Gigabit Ethernet and 40-Gigabit Ethernet uplink ports. If you need an ethernet connection, use GLC-T/GLC-TE copper SFP for one Gigabit Ethernet on all modules.

Note Cisco Catalyst 9300L Series Switches do not support network modules. They only support fixed uplink SFP ports.

The following are the network modules supported on the Cisco Catalyst 9300 Series Switches:

- 4x1G

- 4x10G (Multigigabit Ethernet module)
- 8x10G
- 2x25G
- 2x40G

Cisco Catalyst 9300L Series Switches support only fixed uplink SFP ports of 4x1G and 4x10G.

Multigigabit Ethernet

The MultiGigabit Ethernet (mGig) feature allows you to configure speeds of 100 Mbps, 1 Gbps, 2.5 Gbps, and 5 Gbps with automatic bandwidth negotiation over traditional CAT5e cables and higher cable variants.

The following Cisco Catalyst 9300 series switches support the mGig feature:

- C9300-24UX
- C9300-48UN
- C9300-48UXM

Note Cisco Catalyst 9300L Series Switches do not support Multigigabit Ethernet.

Multigigabit Ethernet supports multi-rate speeds where the ports exchange auto-negotiation pages to establish a link at the highest speed that is supported by both ends of the channel. In a high-noise environment, when port speed downshifting is enabled on an interface, the line rate automatically downgrades to a lower speed when a higher speed link cannot be established or when an established link quality has degraded to a level where the PHY needs to reestablish the link. The following downshift speed values are recommended:

- 10Gbs (downshift to 5Gbs)
- 5Gbs (downshift to 2.5Gbs)
- 2.5Gbs (downshift to 1Gbs)
- 1Gbs (downshift to 100Mbs)

Power over Ethernet

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

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

Note The following SKUs of Cisco Catalyst 9300 Series Switch do not support PoE:

- C9300-24T
- C9300-48T
- C9300-24S
- C9300-48S
- C9300L-24T
- C9300L-48T

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

Using the Switch USB Ports

The device has two USB ports on the front panel: a USB mini-Type B console port and a USB Type A port and a USB 3.0 port on the rear panel.

USB Mini-Type B Console Port

The device has the following console ports:

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

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

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

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

Console Port Change Logs

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

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

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

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

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

USB Type A Port

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

Disabling USB Ports

From Cisco IOS XE Bengaluru 17.5.x, all the USB ports in a standalone or stacked device can be disabled using the **platform usb disable** command. To reenable the USB ports, use the **no platform usb disable** command.

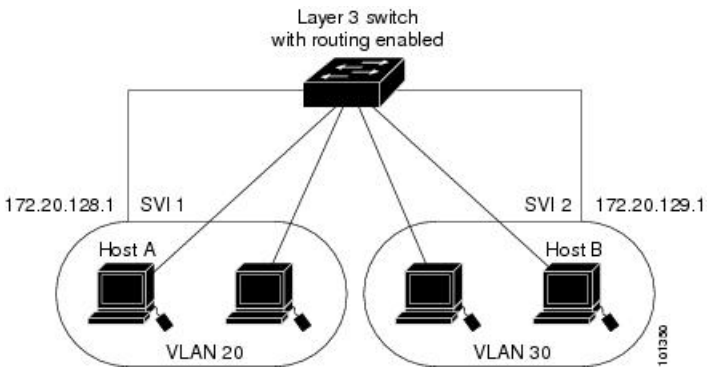
When a USB port is disabled, no system messages are generated if a USB is inserted.

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

Interface Connections

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

Figure 1: Connecting VLANs with a Switch



When the Network Advantage license is used on the device or the active device, the device uses the routing method to forward traffic between interfaces. If the Network Essentials license is used on the device or the active device, only basic routing (static routing and RIP) is supported. Whenever possible, to maintain high performance, forwarding is done by the device hardware. However, only IPv4 packets with Ethernet II encapsulation are routed in hardware.

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

Interface Configuration Mode

The device supports these interface types:

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

You can also configure a range of interfaces.

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

- Type: Gigabit Ethernet (GigabitEthernet or gi) for 10/100/1000 Mbps Ethernet ports, 2.5-Gigabit Ethernet (TwoGigabitEthernet or tw) for 2.5 Gbps, 5-Gigabit Ethernet (FiveGigabitEthernet or fi) for 5 Gbps, 10-Gigabit Ethernet (TenGigabitEthernet or te) for 10 Gbps, 25-Gigabit Ethernet (TwentyFiveGigE or twe) for 25 Gbps, small form-factor pluggable (SFP) module Gigabit Ethernet and 10-Gigabit Ethernet interfaces and quad small-form-factor pluggable (QSFP) module 40-Gigabit Ethernet (FortyGigabitEthernet or fo) for 40 Gbps.

Note On a Cisco Catalyst 9300L Series Switch, the Type can be either Gigabit Ethernet or 10-Gigabit Ethernet.

- Stack member number: The number that identifies the device within the stack. The device number range is 1 to 8 and is assigned the first time the device initializes. The default device number, before it is integrated into a device stack, is 1. When a device has been assigned a stack member number, it keeps that number until another is assigned to it.

  You can use the switch port LEDs in Stack mode to identify the stack member number of a device.

- Module number: The module or slot number on the device: switch (downlink) ports are 0, and uplink ports are 1.

- Port number: The interface number on the device. The 10/100/1000 port numbers always begin at 1, starting with the far left port when facing the front of the device, for example, GigabitEthernet1/0/1 or GigabitEthernet1/0/8.

  On a device with SFP uplink ports, the module number is 1 and the port numbers restart. For example, if the device has 24 10/100/1000 ports, the SFP module ports are GigabitEthernet1/1/1 through GigabitEthernet1/1/4 or TenGigabitEthernet1/1/1 through TenGigabitEthernet1/1/4.

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

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

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

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

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

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

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

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

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

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

Breakout Interfaces

Cisco Catalyst 9300 Series Switches support dual mode breakout cables. Dual mode breakout cables support both 4x10G conversion and straight 40G support. Breakout cables enable a single 40G QSFP+ interface to be split into four 10G SFP+ interfaces.

Breakout cable support is available only on the following switch models and network modules, with a few limitations.

Switch Models

- C9300-24UX
- C9300-48UXM
- C9300-48UN
- C9300L-24UXG-2Q
- C9300L-48UXG-2Q
- C9300X-12Y
- C9300X-24Y

- C9300X-24HX
- C9300X-48HX
- C9300X-48TX
- C9300X-48HXN

Network Modules

- C3850-NM-2-40G
- C9300-NM-2Q
- C9300X-NM-2C
- C9300X-NM-4C

Limitations for Breakout Interfaces

- To enable breakout for dual mode QSFP breakout cables, the **hw-module breakout module** *slot* **port** *port-range* **switch** *switch-num* command must be configured on the two uplink ports of the switch. The range for the variables in the **hw-module breakout module** *slot* **port** *port-range* **switch** *switch-num* command are given below:
  - *slot*: Slot number of port depending on the chassis model. This can be only 1.
  - *port-range*: Single port or range of ports on which breakout is configured. The range is from 1 to 2. On the C9300X-NM-4C network module, the range of ports is from 1 to 4.
  - *switch-num*: Switch number in the stack. The range varies from 1 to 8.

Note On the Cisco Catalyst C9300X-48HXN Series Switches, breakout cable is supported only on port 1 of the C9300X-NM-2C network module. Breakout cable is not supported on port 2 of the C9300X-NM-2C network module. Both port 1 and port 2 can be used for 40/100G SFPs.

See Breakout Interfaces, on page 9 for the list of configurable interfaces.

Default Ethernet Interface Configuration

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

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

Table 1: Default Layer 2 Ethernet Interface Configuration

Feature	Default Setting
Operating mode	Layer 2 or switching mode (**switchport** command).
Allowed VLAN range	VLANs 1 to 4094.
Default VLAN (for access ports)	VLAN 1 (Layer 2 interfaces only).
Native VLAN (for IEEE 802.1Q trunks)	VLAN 1 (Layer 2 interfaces only).
VLAN trunking	Switchport mode dynamic auto (supports DTP) (Layer 2 interfaces only).
Port enable state	All ports are enabled.
Port description	None defined.
Speed	Autonegotiate.(Not supported on the 10-Gigabit interfaces , and also on the fiber SKUs: C9300-24S and C9300-48S.)
Duplex mode	Autonegotiate. (Not supported on the 10-Gigabit interfaces, and also on the fiber SKUs: C9300-24S and C9300-48S.)
Flow control	Flow control is set to **receive: on**. It is always off for sent packets.
EtherChannel (PAgP)	Disabled on all Ethernet ports.
Port blocking (unknown multicast and unknown unicast traffic)	Disabled (not blocked) (Layer 2 interfaces only).
Broadcast, multicast, and unicast storm control	Disabled.
Protected port	Disabled (Layer 2 interfaces only).
Port security	Disabled (Layer 2 interfaces only).
Port Fast	Disabled.
Auto-MDIX	Enabled. **Note** The switch might not support a pre-standard powered device, such as Cisco IP phones and access points that do not fully support IEEE 802.3af, if that powered device is connected to the switch through a crossover cable. This is regardless of whether auto-MIDX is enabled on the switch port.
Power over Ethernet (PoE)	Enabled (auto).(Not supported on C9300-24T, C9300-48T, C9300-24S, and C9300-48S)

Interface Speed and Duplex Mode

Ethernet interfaces on the switch operate at 10, 100, 1000 Mbps, 2.5 Gbps, 5 Gbps, 10 Gbps and in either full-duplex or half-duplex mode. In full-duplex mode, two stations can send and receive traffic at the same time. Normally, 10-Mbps ports operate in half-duplex mode, which means that stations can either receive or send traffic.

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

Note Cisco Catalyst 9300L Series Switches support only SFP uplink ports with speeds up to 1Gbps and SFP+ uplink ports with speeds up to 10 Gbps.

Speed and Duplex Configuration Guidelines

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

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

  Multigigabit ethernet ports (2.5 Gb/s, 5Gb/s, 10 Gb/s) support all speed options but only support auto and full duplex mode. These ports do not support half-duplex mode at any speed.

  SFP ports operating at 1 Gb/s, SFP+ ports operating at 10 Gb/s, SFP28 ports operating at 25 Gb/s and QSFP ports operating at 40 Gb/s only **no speed nonegotiate** or **speed nonegotiate**. Duplex options are not supported.

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

  QSFP ports operating at 40 Gb/s support all speed options but only support auto and full duplex.

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

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

IEEE 802.3x Flow Control

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

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

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

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

These rules apply to flow control settings on the device:

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

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

Layer 3 Interfaces

The device supports these types of Layer 3 interfaces:

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

Note

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

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

- Routed ports: Routed ports are physical ports configured to be in Layer 3 mode by using the **no switchport** interface configuration command. A routed port supports VLAN subinterfaces.

  VLAN subinterface: A 802.1Q VLAN subinterface is a virtual Cisco IOS interface that is associated with a VLAN id on a routed physical interface. The parent interface is a physical port. Subinterfaces can be created only on Layer 3 physical interfaces. A subinterface can be associated with different functionalities such as IP addressing, forwarding policies, Quality of Service (QoS) policies, and security policies. Subinterfaces divide the parent interface into two or more virtual interfaces on which you can assign unique Layer 3 parameters such as IP addresses and dynamic routing protocols. The IP address for each subinterface should be in a different subnet from any other subinterface on the parent interface.

- Layer 3 EtherChannel ports: EtherChannel interfaces made up of routed ports.

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

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

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

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

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

How to Configure Interface Characteristics

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

Configuring an Interface

These general instructions apply to all interface configuration processes.

Procedure

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

Step 3	**interface**
	Example:
	`Device(config)# interface gigabitethernet1/0/1`
	`Device(config-if)#`
		Identifies the interface type, the device number (only on stacking-capable switches), and the number of the connector.
		Note You do not need to add a space between the interface type and the interface number. For example, in the preceding line, you can specify either **gigabitethernet 1/0/1**, **gigabitethernet1/0/1**, **gi 1/0/1**, or **gi1/0/1**.

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

	Command or Action	Purpose
		interface that the device supports or for the specified interface.

Adding a Description for an Interface

Follow these steps to add a description for an interface.

Procedure

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

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

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. • The **macro** variable is explained in Configuring and Using Interface Range Macros. • In a comma-separated *port-range*, you must enter the interface type for each entry and enter spaces before and after the comma. • In a hyphen-separated *port-range*, you do not need to re-enter the interface type, but you must enter a space before the hyphen.

	Command or Action	Purpose
		Note Use the normal configuration commands to apply the configuration parameters to all interfaces in the range. Each command is executed as it is entered.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**show interfaces** [*interface-id*] **Example:** `Device# show interfaces`	Verifies the configuration of the interfaces in the range.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring and Using Interface Range Macros

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**define interface-range** *macro_name interface-range*	Defines the interface-range macro, and saves it in NVRAM.

	Command or Action	Purpose
	Example: `Device(config)# define interface-range enet_list gigabitethernet1/0/1 - 2`	• 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.

Setting the Interface Speed and Duplex Parameters

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet1/0/3`	Specifies the physical interface to be configured, and enters interface configuration mode.
Step 4	**speed** {**10** | **100** | **1000** | **2500** | **5000** | **10000** | **auto** [**10** | **100** | **1000** | **2500** | **5000** | **10000**] | **nonegotiate**} **Example:** `Device(config-if)# speed 10`	Enters the appropriate speed parameter for the interface: • Enter **10**, **100**, **1000**, **2500**, **5000**, or **10000** to set a specific speed for the interface. **Note** Cisco Catalyst 9300L Series Switches support only **10** Mb/s, **100**Mb/s, **1000**Mb/s, **10000** Mb/s, and **auto** speed options. • Enter **auto** to enable the interface to autonegotiate speed with the connected device. If you specify a speed and also set the **auto** keyword, the port autonegotiates only at the specified speeds. • The **nonegotiate** keyword is available only for SFP module ports. SFP module ports operate only at 1000 Mb/s but can be configured to not negotiate if connected to a device that does not support autonegotiation.
Step 5	**duplex** {**auto** | **full** | **half**} **Example:**	Enters the duplex parameter for the interface. Enables half-duplex mode (for interfaces operating only at 10 or 100 Mb/s). Half duplex

	Command or Action	Purpose
	`Device(config-if)# duplex half`	is not supported on multi-Gigabit Ethernet ports configured for speed of 1000 Mb/s. You can configure the duplex setting when the speed is set to **auto**.
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 7	**show interfaces** *interface-id* **Example:** `Device# show interfaces gigabitethernet1/0/3`	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 a Breakout Interface

For information about device compatibility, see the Transceiver Module Group (TMG) Compatibility Matrix.

C9300-NM-2Q Network Module

The default port connections for the C9300-NM-2Q module depends on whether you use a 40G QSFP module or a 4x10G breakout cable.

- If you use a 40G QSFP module, the ports default to 40G interfaces.
- If you use a 4x10G breakout cable, one 40G port is split into four 10G ports.
- You can use a combination of 40G QSFP modules and 4x10G breakout cables.
- For a 40G port —**FortyGigabitEthernet 1/1/***port-num*, the corresponding starting port in every set of the four 10G breakout ports is **TenGigabitEthernet 1/1/***4x**port-num-3*, where *port-num* is the port number. For example, the starting port in the first set of 10G breakout ports is TenGigabitEthernet1/1/1, the starting port in the second set of 10G starting breakout ports is TenGigabitEthernet1/1/5 and so on.

The following tables list all the interfaces which are configurable depending on the type of module and cable used. Note that the **show interface status** command displays all the interfaces in the active state.

- In Table 2: C9300-NM-2Q Module with two 40G QSFP Modules, the 10G interfaces are displayed but are not active.

- In Table 3: C9300-NM-2Q Module with two 4x10G Breakout Cables, the 40G interfaces are displayed but are not active.

Table 2: C9300-NM-2Q Module with two 40G QSFP Modules

Interface	Action
FortyGigabitEthernet1/1/1	Configure this interface
FortyGigabitEthernet1/1/2	Configure this interface
TenGigabitEthernet1/1/1	Disregard
TenGigabitEthernet1/1/2	Disregard
TenGigabitEthernet1/1/3	Disregard
TenGigabitEthernet1/1/4	Disregard
TenGigabitEthernet1/1/5	Disregard
TenGigabitEthernet1/1/6	Disregard
TenGigabitEthernet1/1/7	Disregard
TenGigabitEthernet1/1/8	Disregard

Table 3: C9300-NM-2Q Module with two 4x10G Breakout Cables

Interface	Action
FortyGigabitEthernet1/1/1	Disregard
FortyGigabitEthernet1/1/2	Disregard
TenGigabitEthernet1/1/1	Configure this interface
TenGigabitEthernet1/1/2	Configure this interface
TenGigabitEthernet1/1/3	Configure this interface
TenGigabitEthernet1/1/4	Configure this interface
TenGigabitEthernet1/1/5	Configure this interface
TenGigabitEthernet1/1/6	Configure this interface
TenGigabitEthernet1/1/7	Configure this interface
TenGigabitEthernet1/1/8	Configure this interface

Configuring Forty Gigabit Ethernet Interface

Follow these steps to configure the forty gigabit ethernet interface. Use the no form of the command to disable the fortygigabit ethernet 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 fortygigabitethernet1/0/9`
	`Device(config-if)#`
		Specifies the interface type,that has to be configured.

Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring the IEEE 802.3x Flow Control

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

Procedure

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

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet1/0/1`	Specifies the physical interface to be configured, and enters interface configuration mode.
Step 4	**flowcontrol** {**receive**} {**on** | **off** | **desired**} **Example:** `Device(config-if)# flowcontrol receive on`	Configures the flow control mode for the port.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show interfaces** *interface-id* **Example:** `Device# show interfaces gigabitethernet1/0/1`	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.

Configuring a Layer 3 Interface

Follow these steps to configure a layer 3 interface.

Procedure

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

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

Configuring a Logical Layer 3 GRE Tunnel Interface

Before you begin

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

Note

- GRE tunnels are supported on the hardware on Cisco Catalyst 9000 switches. When GRE is configured without tunnel options, packets are hardware-switched. When GRE is configured with tunnel options (such as key, checksum, and so on), packets are switched in the software. A maximum of 100 GRE tunnels are supported.
- Other features such as Access Control Lists (ACL) and Quality of Service (QoS) are not supported for the GRE tunnels.
- The **tunnel path-mtu-discovery** command is not supported for GRE tunnels. To avoid fragmentation, you can set the maximum transmission unit (MTU) of both ends of the GRE tunnel to the lowest value by using the **ip mtu 256** command.

To configure a GRE tunnel, perform this task:

Procedure

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

	Command or Action	Purpose
	`Device(config)#tunnel source 10.10.10.1`	
Step 6	**tunnel destination** {*host_name* | *ip_address*} **Example:** `Device(config)#tunnel destination 10.10.10.2`	Configures the tunnel destination.
Step 7	**tunnel mode gre ip** **Example:** `Device(config)#tunnel mode gre ip`	Configures the tunnel mode.
Step 8	**end** **Example:** `Device(config)#end`	Exits configuration mode.

Configuring SVI Autostate Exclude

Follow these steps to exclude SVI autostate.

Procedure

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

	Command or Action	Purpose
	exclude	
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show running config interface** *interface-id*	(Optional) Shows the running configuration. Verifies the configuration.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Shutting Down and Restarting an Interface

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

Procedure

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

	Command or Action	Purpose
	`gigabitethernet1/0/2`	
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.

This configuration applies to all switches in a stack.

Procedure

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

	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 switch** *switch_number* **Example:** `Device(config-line)# media-type rj45 switch 1`	Configures the console media type to be only RJ-45 port. If you do not enter this command and both types are connected, the USB port is used by default.
Step 5	**end** **Example:** `Device(config)# 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 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 The configured inactivity timeout applies to all device in a stack. However, a timeout on one device does not cause a timeout on other device in the stack.

Procedure

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

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

Disabling USB Ports

To disable all USB ports, peform this procedure.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	[**no**] **platform usb disable** **Example:**	Disables all the USB ports on the device.

	Command or Action	Purpose
	`Device(config)# platform usb disable`	Use the **no platform usb disable** command to reenable the USB ports.
Step 4	**exit** **Example:** `Device(config)# exit`	Exits to privileged EXEC mode.
Step 5	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Monitoring Interface Characteristics

The following sections provide information about monitoring interface characteristics.

Monitoring Interface Status

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

Table 4: show Commands for Interfaces

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

Command	Purpose
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 5: clear Commands for Interfaces

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

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

Configuration Examples for Interface Characteristics

The following sections provide examples of interface characteristics configurations.

Example: Adding a Description to an Interface

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

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

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

Example: Configuring Interfaces on a Stack-Capable Switch

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

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

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

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

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

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

Example: Configuring a Range of Interfaces

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

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

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

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

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

Example: Configuring and Using Interface Range Macros

The following example shows how to 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 gigabitethernet1/0/1 - 2
Device(config)# end
Device# show running-config | include define
define interface-range enet_list GigabitEthernet1/0/1 - 2
```

The following example shows how to create a multiple-interface macro named *macro1*:

```
Device# configure terminal
Device(config)# define interface-range macro1 gigabitethernet1/0/1 - 2, gigabitethernet1/0/5
 - 7, tengigabitethernet1/1/1 -2
Device(config)# end
```

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

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

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

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

Example: Setting Interface Speed and Duplex Mode

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

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

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

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

Example: Configuring a Layer 3 Interface

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

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
Device(config)# interface gigabitethernet1/0/2
Device(config-if)# no switchport
Device(config-if)# ip address 192.20.135.21 255.255.255.0
Device(config-if)# no shutdown
```

Example: Configuring a Breakout Interface

The following example shows a sample output of the **show interface status** command with dual mode 40G QSFP module inserted into port number 2:

```
Device# show interface status

Port      Name               Status       Vlan       Duplex  Speed Type
Fo2/0/1                      notconnect   1            auto   auto unknown
Fo2/0/2                       notconnect   1             full    40G QSFP
40G SR4 SFP
Fo2/0/3                      notconnect   1            auto   auto unknown
Fo2/0/4                      notconnect   1            auto   auto unknown
Fo2/0/5                      notconnect   1            auto   auto unknown
Fo2/0/6                      notconnect   1            auto   auto unknown
Fo2/0/7                      notconnect   1            auto   auto unknown
Fo2/0/8                      notconnect   1            auto   auto unknown
Fo2/0/9                      notconnect   1            auto   auto unknown
Fo2/0/10                     notconnect   1            auto   auto unknown
Fo2/0/11                     notconnect   1            auto   auto unknown
Fo2/0/12                     notconnect   1            auto   auto unknown
Fo2/0/13                     notconnect   1            auto   auto unknown
Fo2/0/14                     notconnect   1            auto   auto unknown
Fo2/0/15                     notconnect   1            auto   auto unknown
Fo2/0/16                     notconnect   1            auto   auto unknown
Fo2/0/17                     notconnect   1            auto   auto unknown
Fo2/0/18                     notconnect   1            auto   auto unknown
Fo2/0/19                     notconnect   1            auto   auto unknown
Fo2/0/20                     notconnect   1            auto   auto unknown
Fo2/0/21                     notconnect   1            auto   auto unknown
Fo2/0/22                     notconnect   1            auto   auto unknown
Fo2/0/23                     notconnect   1            auto   auto unknown
Fo2/0/24                     notconnect   1            auto   auto unknown
…….……..……..……..
```

```
*May 17 21:35:27.399 UTC: %PLATFORM_PM-6-FRULINK_REMOVED: 1x40G Port2
uplink module removed from switch 2 slot 1
*May 17 21:35:27.899 UTC: %PLATFORM_PM-6-FRULINK_INSERTED: BC:4x10G Port2
 uplink module inserted in the switch 2 slot 1
*May 17 21:35:29.399 UTC: %LINK-3-UPDOWN: Interface
FortyGigabitEthernet2/0/2, changed state to down
*May 17 21:35:31.181 UTC: %PLATFORM_PM-6-MODULE_INSERTED: SFP module
inserted with interface name Te2/0/5
*May 17 21:35:33.414 UTC: %PLATFORM_PM-6-MODULE_INSERTED: SFP module
inserted with interface name Te2/0/6
*May 17 21:35:35.648 UTC: %PLATFORM_PM-6-MODULE_INSERTED: SFP module
inserted with interface name Te2/0/7
*May 17 21:35:37.881 UTC: %PLATFORM_PM-6-MODULE_INSERTED: SFP module
inserted with interface name Te2/0/8
*May 17 21:35:42.234 UTC: %LINK-3-UPDOWN: Interface
TenGigabitEthernet2/0/5, changed state to up
*May 17 21:35:43.234 UTC: %LINEPROTO-5-UPDOWN: Line protocol on Interface
 TenGigabitEthernet2/0/5, changed state to up
*May 17 21:35:51.460 UTC: %LINK-3-UPDOWN: Interface
TenGigabitEthernet2/0/6, changed state to up
*May 17 21:35:51.506 UTC: %LINK-3-UPDOWN: Interface
TenGigabitEthernet2/0/7, changed state to up
*May 17 21:35:51.551 UTC: %LINK-3-UPDOWN: Interface
TenGigabitEthernet2/0/8, changed state to up
*May 17 21:35:52.286 UTC: %LINEPROTO-5-UPDOWN: Line protocol on Interface
 Vlan1, changed state to up
*May 17 21:35:52.461 UTC: %LINEPROTO-5-UPDOWN: Line protocol on Interface
 TenGigabitEthernet2/0/6, changed state to up
*May 17 21:35:52.505 UTC: %LINEPROTO-5-UPDOWN: Line protocol on Interface
 TenGigabitEthernet2/0/7, changed state to up
*May 17 21:35:52.551 UTC: %LINEPROTO-5-UPDOWN: Line protocol on Interface
 TenGigabitEthernet2/0/8, changed state to up
Device(config)# end
Device# show interface status

Port      Name               Status       Vlan       Duplex  Speed Type
Fo2/0/1                      notconnect   1            auto   auto unknown
Fo2/0/3                      notconnect   1            auto   auto unknown
Fo2/0/4                      notconnect   1            auto   auto unknown
Fo2/0/5                      notconnect   1            auto   auto unknown
Fo2/0/6                      notconnect   1            auto   auto unknown
Fo2/0/7                      notconnect   1            auto   auto unknown
Fo2/0/8                      notconnect   1            auto   auto unknown
Fo2/0/9                      notconnect   1            auto   auto unknown
Fo2/0/10                     notconnect   1            auto   auto unknown
Fo2/0/11                     notconnect   1            auto   auto unknown
Fo2/0/12                     notconnect   1            auto   auto unknown
Fo2/0/13                     notconnect   1            auto   auto unknown
Fo2/0/14                     notconnect   1            auto   auto unknown
Fo2/0/15                     notconnect   1            auto   auto unknown
Fo2/0/16                     notconnect   1            auto   auto unknown
Fo2/0/17                     notconnect   1            auto   auto unknown
```

```
Fo2/0/18                    notconnect   1            auto   auto unknown
Fo2/0/19                    notconnect   1            auto   auto unknown
Fo2/0/20                    notconnect   1            auto   auto unknown
Fo2/0/21                    notconnect   1            auto   auto unknown
Fo2/0/22                    notconnect   1            auto   auto unknown
Fo2/0/23                    notconnect   1            auto   auto unknown
Fo2/0/24                    notconnect   1            auto   auto unknown
…….……..……..……..
```

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

The following example shows how to disable the configuration:

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

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

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

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

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

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

Additional References for Configuring Interface Characteristics

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the "Interface and Hardware Commands" section in the *Command Reference (Catalyst 9300 Series Switches)* .

Feature History for Configuring Interface Characteristics

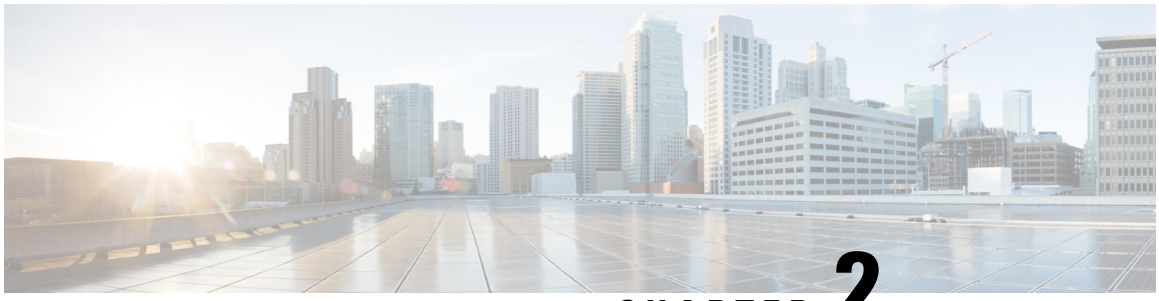
This table provides release and related information for the features explained in this module.

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

Release	Feature	Feature Information
Cisco IOS XE Everest 16.5.1a	Interface Characteristics	Interface Characteristics includes interface types, connections, configuration modes, speed, and other aspects of configuring a physical interface on a device. Support for this feature was introduced only on the 9300 switch models of the Cisco Catalyst 9300 Series Switches.
Cisco IOS XE Everest 16.6.4	IEEE 802.3x Flow Control	The default value for **flowcontrol** interface configuration command was modified to **on** on all the models of the series.
Cisco IOS XE Fuji 16.8.1a	Breakout interfaces	Support for breakout interfaces was introduced on the following: • Only the first four ports of C9300-24UX, C9300-48UXM and C9300-48UN models. • All the ports of the C9300-NM-2Q network module support breakout configuration
Cisco IOS XE Fuji 16.9.1	Breakout interfaces	On Cisco Catalyst 9300 Series Switches, support for breakout configuration was introduced only on the first twelve ports of C9300-24UX, C9300-48UXM and C9300-48UN models.
Cisco IOS XE Gibraltar 16.10.1	Password Authentication on USB 3.0 SSD	Support for configuring password on a USB 3.0 SSD was enabled on all the models of the Cisco Catalyst 9300 Series Switches.
Cisco IOS XE Gibraltar 16.11.1c	Interface Characteristics	Support for configuration of interface characteristics was introduced on the 9300L switch models of the Cisco Catalyst 9300 Series Switches.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.12.2	Breakout interfaces	Support for breakout configuration was introduced on the C9300L-24UXG-2Q and C9300L-48UXG-2Q models of the Cisco Catalyst 9300L Series Switches.
Cisco IOS XE Bengaluru 17.5.1	Disabling USB interfaces	Support to disable all USB ports on a standalone or stacked device was introduced.

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



CHAPTER 2

Configuring Auto-MDIX

- Prerequisites for Auto-MDIX, on page 43
- Restrictions for Auto-MDIX, on page 43
- Information About Configuring Auto-MDIX, on page 44
- How to Configure Auto-MDIX, on page 44
- Example for Configuring Auto-MDIX, on page 45
- Auto-MDIX and Operational State, on page 46
- Additional References for Auto-MDIX, on page 46
- Feature History for Auto-MDIX, on page 46

Prerequisites for Auto-MDIX

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

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

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 any other SFP, SFP+ , or QSFP module interface.

Restrictions for Auto-MDIX

The device might not support a pre-standard powered device—such as Cisco IP phones and access points that do not fully support IEEE 802.3af—if that powered device is connected to the device through a crossover cable. This is regardless of whether auto-MIDX is enabled on the switch port.

Information About Configuring Auto-MDIX

Auto-MDIX on an Interface

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

Note Auto-MDIX is enabled by default.

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

Table 6: Link Conditions and Auto-MDIX Settings

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

How to Configure Auto-MDIX

Configuring Auto-MDIX on an Interface

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

Procedure

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet1/0/1`	Specifies the physical interface to be configured, and enter interface configuration mode.
Step 4	**mdix auto** **Example:** `Device(config-if)# mdix auto`	Enables the Auto MDIX feature.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Example for Configuring Auto-MDIX

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

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

Auto-MDIX and Operational State

Table 7: Auto-MDIX and Operational State

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

Additional References for Auto-MDIX

Related Documentation

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	*Command Reference (Catalyst 9300 Series Switches)*
For information about the power supplies.	*Cisco Catalyst 9300 Series Switches Hardware Installation Guide*

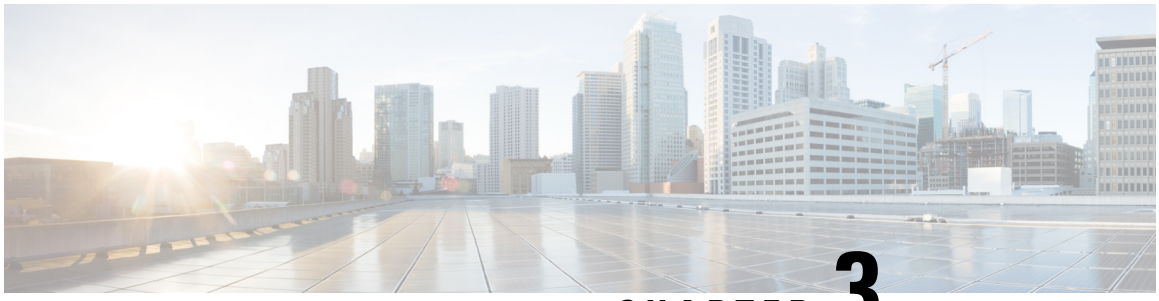
Feature History for Auto-MDIX

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

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

Release	Feature	Feature Information
Cisco IOS XE Everest 16.5.1a	Auto-MDIX on an Interface	An automatic medium-dependent interface crossover (Auto-MDIX) enabled interface detects the required cable connection type (straight through or crossover) and configures the connection appropriately.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 3

Configuring Ethernet Management Port

- Prerequisites for Ethernet Management Port, on page 49
- Information About the Ethernet Management Port, on page 49
- How to Configure the Ethernet Management Port, on page 52
- Example for Configuring IP Address on Ethernet Management Interface, on page 53
- Additional References for Ethernet Management Port, on page 53
- Feature History for Ethernet Management Port, on page 54

Prerequisites for Ethernet Management Port

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

Information About the Ethernet Management Port

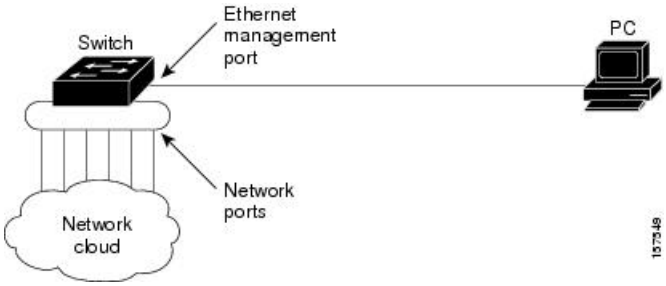
The Ethernet management port, also referred to as the *Gi0/0* or *GigabitEthernet0/0* port, is a VRF (VPN routing/forwarding) interface to which you can connect a PC. You can use the Ethernet management port instead of the device console port for network management.

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

Ethernet Management Port Direct Connection to a Device

Figure 2: Connecting a Device to a PC

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

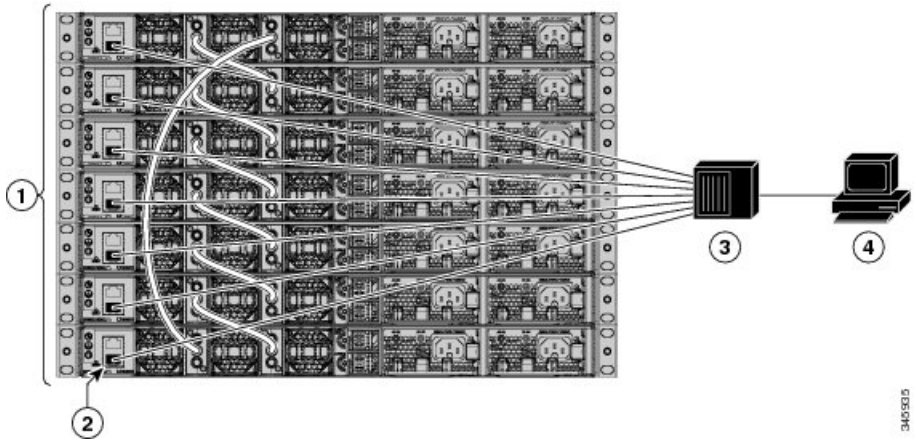


Ethernet Management Port Connection to Stack Devices using a Hub

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

Figure 3: Connecting a Device Stack to a PC

This figure displays how a PC uses a hub to connect to a device stack.



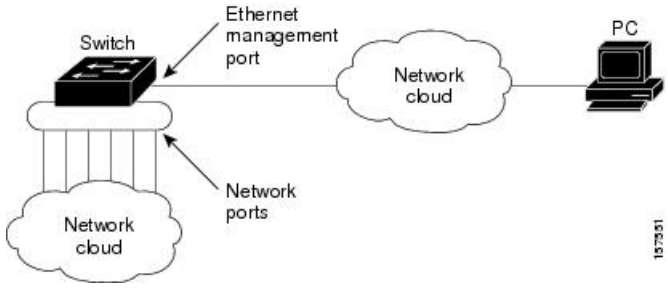
1	Switch stack	3	Hub
2	Management port	4	PC

Ethernet Management Port and Routing

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

Figure 4: Network Example with Routing Protocols Enabled

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



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

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

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

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

Supported Features on the Ethernet Management Port

The Ethernet management port supports these features:

- Express Setup (only in device stacks)
- Network Assistant
- Telnet with passwords
- TFTP
- Secure Shell (SSH)
- DHCP-based autoconfiguration
- SNMP (IF-MIB)
- IP ping
- Interface features:
  - Speed: 10 Mb/s, 100 Mb/s, 1000 Mb/s, and autonegotiation
  - Duplex mode: Full, half, and autonegotiation
  - Loopback detection
- Cisco Discovery Protocol (CDP)
- DHCP relay agent

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

How to Configure the Ethernet Management Port

Disabling and Enabling the Ethernet Management Port

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**interface gigabitethernet0/0** **Example:** `Device(config)# interface gigabitethernet0/0`	Specifies the Ethernet management port in the CLI.
Step 3	**shutdown** **Example:** `Device(config-if)# shutdown`	Disables the Ethernet management port.
Step 4	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Enables the Ethernet management port.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 6	**show interfaces gigabitethernet0/0** **Example:** `Device# show interfaces gigabitethernet0/0`	Displays the link status. To find out the link status to the PC, you can monitor the LED for the Ethernet management port. The LED is green (on) when the link is active, and the LED is off when the link is down. The LED is amber when there is a POST failure.

What to do next

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

Example for Configuring IP Address on Ethernet Management Interface

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

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


Device# show running-config interface Gi0/0
Building configuration...

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

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

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


Device#show running-config interface Te0/1
Building configuration...

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

Additional References for Ethernet Management Port

Related Documents

Related Topic	Document Title
Bootloader configuration	See the *System Management* section of this guide.
Bootloader commands	See the *System Management Commands* section of the *Command Reference (Catalyst 9300 Series Switches)*

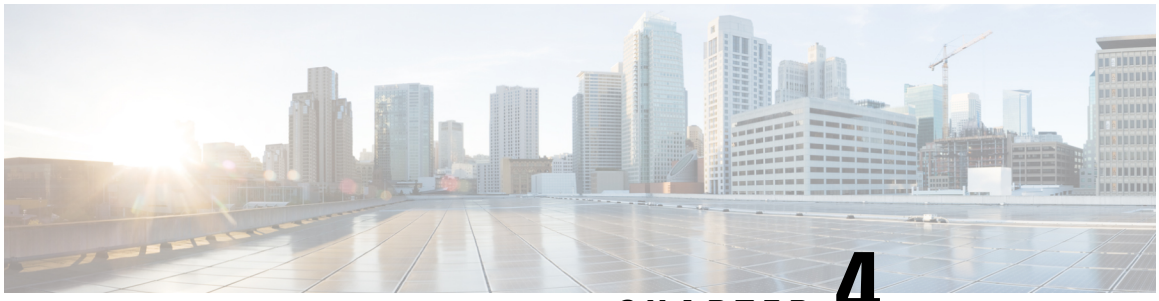
Feature History for Ethernet Management Port

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

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

Release	Feature	Feature Information
Cisco IOS XE Everest 16.5.1a	Ethernet Management Port	The Ethernet management port is a VRF interface to which you can connect a PC. You can use the Ethernet management port instead of the device console port for network management.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 4

Checking Port Status and Connectivity

- Check Cable Status Using Time Domain Reflectometer, on page 55
- Feature History for Checking Port Status and Connectivity, on page 57

Check Cable Status Using Time Domain Reflectometer

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

With TDR, you can check the status of copper cables for the ports on the Catalyst 9300 Series Switches. TDR detects a cable fault by sending a signal through the cable and reading the signal that is reflected back. All or part of the signal can be reflected back due to defects in the cable.

Note Category 5 cable has four pairs. Each pair can assume one of the following states: open (not connected), broken, shorted, or terminated. The TDR test detects all four states and displays the first three as “Fault” conditions, and displays the fourth as “Normal”.

TDR feature is supported on the following modules:

- C9300-24T
- C9300-48T
- C9300-24P
- C9300-48P
- C9300-24U
- C9300-48U
- C9300-24UX
- C9300-48UXM
- C9300-48UN

TDR detects a cable fault by sending a signal along its wires. Depending on the reflected signal, it can determine roughly where a cable fault could be. The variations on how TDR signal is reflected back determine the results

on TDR. On Catalyst 9300 Series Switches, only two types of cable fault types are detected - OPEN, SHORT and IMPEDANCE MISMATCH. We do display Normal status in case cable is properly terminated and this is done for illustrative purpose.

Running the TDR Test

To start the TDR test, perform this task:

Procedure

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

TDR Guidelines

The following guidelines apply to the use of TDR:

- Do not change the port configuration while the TDR test is running.
- If you connect a port undergoing a TDR test to an Auto-MDIX enabled port, the TDR result might be invalid.
- If you connect a port undergoing a TDR test to a 100BASE-T port such as that on the device, the unused pairs (4-5 and 7-8) are reported as faulty because the remote end does not terminate these pairs.
- Due to cable characteristics, you should run the TDR test multiple times to get accurate results.
- Do not change port status (for example, remove the cable at the near or far end) because the results might be inaccurate.
- TDR works best if the test cable is disconnected from the remote port. Otherwise, it might be difficult for you to interpret results correctly.
- TDR operates across four wires. Depending on the cable conditions, the status might show that one pair is OPEN or SHORT while all other wire pairs display as faulty. This operation is acceptable because you should declare a cable faulty provided one pair of wires is either OPEN or SHORT.
- TDR intent is to determine how poorly a cable is functioning rather than to locate a faulty cable.
- When TDR locates a faulty cable, you should still use an offline cable diagnosis tool to better diagnose the problem.
- TDR results might differ between runs on different switch models of Catalyst 9300 Series Switches because of the resolution difference of TDR implementations. When this occurs, you should refer to an offline cable diagnosis tool.

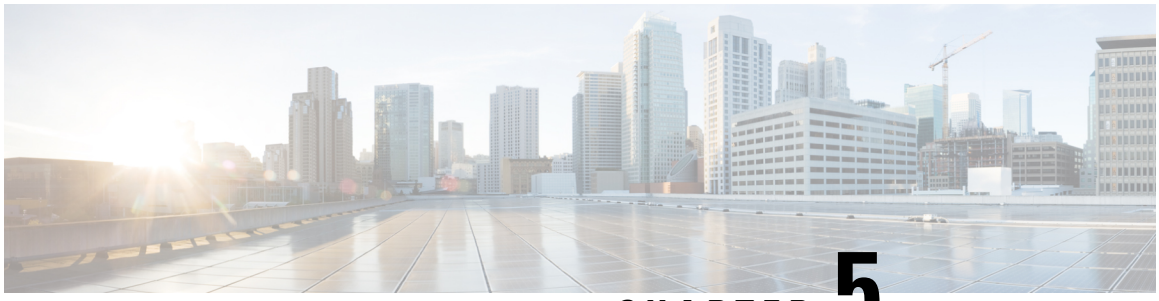
Feature History for Checking Port Status and Connectivity

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

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

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

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 5

Configuring LLDP, LLDP-MED, and Wired Location Service

- Restrictions for LLDP, on page 59
- Information About LLDP, LLDP-MED, and Wired Location Service, on page 59
- How to Configure LLDP, LLDP-MED, and Wired Location Service, on page 63
- Configuration Examples for LLDP, LLDP-MED, and Wired Location Service, on page 73
- Monitoring and Maintaining LLDP, LLDP-MED, and Wired Location Service, on page 74
- Additional References for LLDP, LLDP-MED, and Wired Location Service, on page 75
- Feature History for LLDP, LLDP-MED, and Wired Location Service, on page 75

Restrictions for LLDP

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

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

LLDP

The Cisco Discovery Protocol (CDP) is a device discovery protocol that runs over Layer 2 (the data link layer) on all Cisco-manufactured devices (routers, bridges, access servers, switches, and controllers). CDP allows

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

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

LLDP Supported TLVs

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

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

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

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

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

LLDP-MED

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

LLDP-MED Supported TLVs

LLDP-MED supports these TLVs:

- LLDP-MED capabilities TLV

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

- Network policy TLV

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

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

- Power management TLV

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

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

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

  - Geographic location information

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

  - custom location

    Provides customized name and value of a switch location.

Wired Location Service

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

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

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

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

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

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

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

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

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

Default LLDP Configuration

Table 8: Default LLDP Configuration

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

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

Enabling LLDP

Procedure

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

	Command or Action	Purpose
Step 3	**lldp run** **Example:** `Device(config)# lldp run`	Enables LLDP globally on the device.
Step 4	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet2/0/1`	Specifies the interface on which you are enabling LLDP, and enter interface configuration mode.
Step 5	**lldp transmit** **Example:** `Device(config-if)# lldp transmit`	Enables the interface to send LLDP packets.
Step 6	**lldp receive** **Example:** `Device(config-if)# lldp receive`	Enables the interface to receive LLDP packets.
Step 7	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 8	**show lldp** **Example:** `Device# show lldp`	Verifies the configuration.
Step 9	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring LLDP Characteristics

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

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**lldp holdtime** *seconds* **Example:** `Device(config)# lldp holdtime 120`	(Optional) Specifies the amount of time a receiving device should hold the information from your device before discarding it. The range is 0 to 65535 seconds; the default is 120 seconds.
Step 4	**lldp reinit** *delay* **Example:** `Device(config)# lldp reinit 2`	(Optional) Specifies the delay time in seconds for LLDP to initialize on an interface. The range is 2 to 5 seconds; the default is 2 seconds.
Step 5	**lldp timer** *rate* **Example:** `Device(config)# lldp timer 30`	(Optional) Sets the sending frequency of LLDP updates in seconds. The range is 5 to 65534 seconds; the default is 30 seconds.
Step 6	**lldp tlv-select** **Example:** `Device(config)# tlv-select`	(Optional) Specifies the LLDP TLVs to send or receive.
Step 7	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet2/0/1`	Specifies the interface on which you are enabling LLDP, and enter interface configuration mode.

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

Configuring LLDP-MED TLVs

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

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

Table 9: LLDP-MED TLVs

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

Follow these steps to enable a TLV on an interface:

Procedure

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

Configuring Network-Policy TLV

Procedure

	Command or Action	Purpose
Step 1	**enable** **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).

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

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

Configuring Location TLV and Wired Location Service

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

Procedure

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

Step 2	**location** {**admin-tag** *string* | **civic-location identifier** {*id* | **host**} | **elin-location** *string* **identifier** *id* | **custom-location identifier** {*id* | **host**} | **geo-location identifier** {*id* | **host**}}
	Example:
	`Device(config)# location civic-location identifier 1`
	`Device(config-civic)# number 3550`
	`Device(config-civic)# primary-road-name "Cisco Way"`
	`Device(config-civic)# city "San Jose"`
	`Device(config-civic)# state CA`
	`Device(config-civic)# building 19`
	`Device(config-civic)# room C6`
	`Device(config-civic)# county "Santa Clara"`
	`Device(config-civic)# country US`
		Specifies the location information for an endpoint.
		• **admin-tag**—Specifies an administrative tag or site information.
		• **civic-location**—Specifies civic location information.
		• **elin-location**—Specifies emergency location information (ELIN).
		• **custom-location**—Specifies custom location information.
		• **geo-location**—Specifies geo-spatial location information.
		• **identifier** *id*—Specifies the ID for the civic, ELIN, custom, or geo location.
		• **host**—Specifies the host civic, custom, or geo location.
		• *string*—Specifies the site or location information in alphanumeric format.

Step 3	**exit** **Example:** `Device(config-civic)# exit`	Returns to global configuration mode.

	Command or Action	Purpose
Step 4	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet2/0/1`	Specifies the interface on which you are configuring the location information, and enter interface configuration mode.
Step 5	**location** {**additional-location-information** *word* | **civic-location-id** {*id* | **host**} | **elin-location-id** *id* | **custom-location-id** {*id* | **host**} | **geo-location-id** {*id* | **host**} } **Example:** `Device(config-if)# location elin-location-id 1`	Enters location information for an interface: • **additional-location-information**—Specifies additional information for a location or place. • **civic-location-id**—Specifies global civic location information for an interface. • **elin-location-id**—Specifies emergency location information for an interface. • **custom-location-id**—Specifies custom location information for an interface. • **geo-location-id**—Specifies geo-spatial location information for an interface. • **host**—Specifies the host location identifier. • *word*—Specifies a word or phrase with additional location information. • *id*—Specifies the ID for the civic, ELIN, custom, or geo location. The ID range is 1 to 4095.
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 7	Use one of the following: • **show location admin-tag** *string* • **show location civic-location identifier** *id* • **show location elin-location identifier** *id* **Example:** `Device# show location admin-tag` or	Verifies the configuration.

	Command or Action	Purpose

	`Device# show location civic-location identifier`
	
	or
	
	`Device# show location elin-location identifier`	

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

Enabling Wired Location Service on the Device

Before you begin

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**nmsp notification interval** {**attachment** | **location**} *interval-seconds* **Example:** `Device(config)# nmsp notification interval location 10`	Specifies the NMSP notification interval. **attachment**—Specifies the attachment notification interval. **location**—Specifies the location notification interval. *interval-seconds*—Duration in seconds before the device sends the MSE the location or

	Command or Action	Purpose
		attachment updates. The range is 1 to 30; the default is 30.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**show network-policy profile** **Example:** `Device# show network-policy profile`	Verifies the configuration.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

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

Configuring Network-Policy TLV: Examples

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

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

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

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

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

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

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

Command	Description
clear lldp counters	Resets the traffic counters to zero.
clear lldp table	Deletes the LLDP neighbor information table.
clear nmsp statistics	Clears the NMSP statistic counters.
show lldp	Displays global information, such as frequency of transmissions, the holdtime for packets being sent, and the delay time before LLDP initializes on an interface.
show lldp entry *entry-name*	Displays information about a specific neighbor. You can enter an asterisk (*) to display all neighbors, or you can enter the neighbor name.
show lldp interface [*interface-id*]	Displays information about interfaces with LLDP enabled. You can limit the display to a specific interface.
show lldp neighbors [*interface-id*] [**detail**]	Displays information about neighbors, including device type, interface type and number, holdtime settings, capabilities, and port ID. You can limit the display to neighbors of a specific interface or expand the display for more detailed information.
show lldp traffic	Displays LLDP counters, including the number of packets sent and received, number of packets discarded, and number of unrecognized TLVs.
show location admin-tag *string*	Displays the location information for the specified administrative tag or site.
show location civic-location identifier *id*	Displays the location information for a specific global civic location.
show location elin-location identifier *id*	Displays the location information for an emergency location
show network-policy profile	Displays the configured network-policy profiles.
show nmsp	Displays the NMSP information

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

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the *Interface and Hardware Commands* section in the *Command Reference (Catalyst 9300 Series Switches)*

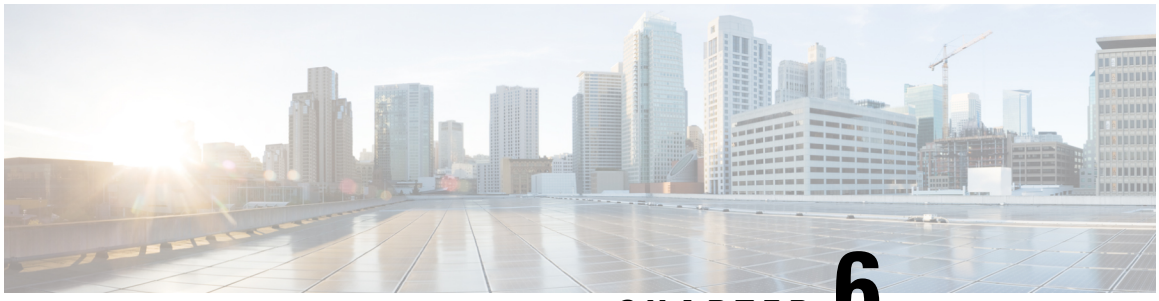
Feature History for LLDP, LLDP-MED, and Wired Location Service

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

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

Release	Feature	Feature Information
Cisco IOS XE Everest 16.5.1a	Link Layer Discovery Protocol (LLDP), LLDP-MED, Wired Location Service	LLDP is a neighbor discovery protocol that is used for network devices to advertise information about themselves to other devices on the network. This protocol runs over the data-link layer, which allows two systems running different network layer protocols to learn about each other. LLDP-MED operates between endpoints and network devices. Wired Location Service lets you send tracking information of the connected devices to a Cisco Mobility Services Engine (MSE).

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 6

Configuring System MTU

- Restrictions for System MTU, on page 77
- Information About the MTU, on page 77
- How to Configure MTU , on page 78
- Configuration Examples for System MTU, on page 79
- Additional References for System MTU, on page 80
- Feature History for System MTU, on page 80

Restrictions for System MTU

Information About the MTU

The default maximum transmission unit (MTU) size for payload received in Ethernet frame and sent on all device interfaces is 1500 bytes. The maximum value of System MTU is 9198 bytes.

System MTU Value Application

This table shows how the MTU values are applied.

Table 10: MTU Values

Configuration	system mtu command	ip mtu command	ipv6 mtu command
Standalone switch or switch stack	You can enter the **system mtu** command on a switch or switch stack. It affects all ports. The range is from 1500 to 9198 bytes.	Use the **ip mtu** *bytes* command. The range is from 832 up to 1500 bytes. **Note** The IP MTU value is the applied value, not the configured value.	Use the **ipv6 mtu** *bytes* command. The range is from 1280 to the system jumbo MTU value (in bytes). **Note** The IPv6 MTU value is the applied value, not the configured value.

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

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

How to Configure MTU

Configuring the System MTU

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**system mtu** *bytes* **Example:** `Device(config)# system mtu 1900`	(Optional) Changes the MTU size for all Gigabit Ethernet and 10-Gigabit Ethernet interfaces.
Step 4	**end** **Example:** `Device(config)#  end`	Returns to privileged EXEC mode.
Step 5	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves your entries in the configuration file.
Step 6	**show system mtu** **Example:** `Device# show system mtu`	Verifies your settings.

Configuring Protocol-Specific MTU

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

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**interface** *interface* **Example:** `Device(config)# interface gigabitethernet0/0`	Enters interface configuration mode.
Step 3	**ip mtu** *bytes* **Example:** `Device(config-if)# ip mtu 68`	Changes the IPv4 MTU size
Step 4	**ipv6 mtu** *bytes* **Example:** `Device(config-if)# ipv6 mtu 1280`	(Optional) Changes the IPv6 MTU size.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves your entries in the configuration file.
Step 7	**show system mtu** **Example:** `Device# show system mtu`	Verifies your settings.

Configuration Examples for System MTU

Example: Configuring Protocol-Specific MTU

```
Device# configure terminal
Device(config)# interface gigabitethernet 0/1
```

```
Device(config-if)# ip mtu 900
Device(config-if)# ipv6 mtu 1286
Device(config-if)# end
```

Example: Configuring the System MTU

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

Additional References for System MTU

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the *Interface and Hardware Commands* section in the *Command Reference (Catalyst 9300 Series Switches)*

Standards and RFCs

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

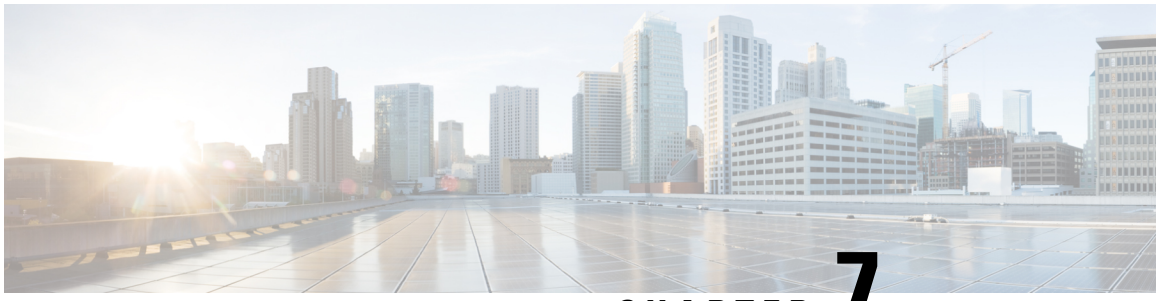
Feature History for System MTU

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

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

Release	Feature	Feature Information
Cisco IOS XE Everest 16.5.1a	System MTU	System MTU defines the maximum transmission unit size for frames transmitted on all interfaces of a switch.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 7

Configuring Per-Port MTU

- Restrictions for Per-Port MTU, on page 81
- Information About Per-Port MTU, on page 81
- Configuring Per-Port MTU, on page 82
- Example: Configuring Per-Port MTU, on page 82
- Example: Verifying Per-Port MTU, on page 83
- Example: Disabling Per-Port MTU, on page 83
- Feature History for Per-Port MTU, on page 83

Restrictions for Per-Port MTU

- Per-Port MTU cannot be configured on the management port.
- Per-Port MTU cannot be configured on SVL links.
- Members of a port channel cannot be configured with Per-Port MTU, they derive their MTU from the port-channel MTU configuration.
- Per-Port MTU is not supported on sub interfaces and port-channel sub interfaces.

Information About Per-Port MTU

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

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

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

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

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

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

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

Configuring Per-Port MTU

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

Procedure

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

Example: Configuring Per-Port MTU

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

```
Device# configure terminal
Device(config)# interface FortyGigabitEthernet2/5/0/20
Device(config-if)# mtu 6666
```

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

Example: Verifying Per-Port MTU

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

```
Device# show interface mtu
Port           Name               MTU
Fo2/5/0/19                        1500
Fo2/5/0/20                        6666
Fo2/5/0/21      ixia_7_21         1500
```

Example: Disabling Per-Port MTU

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

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

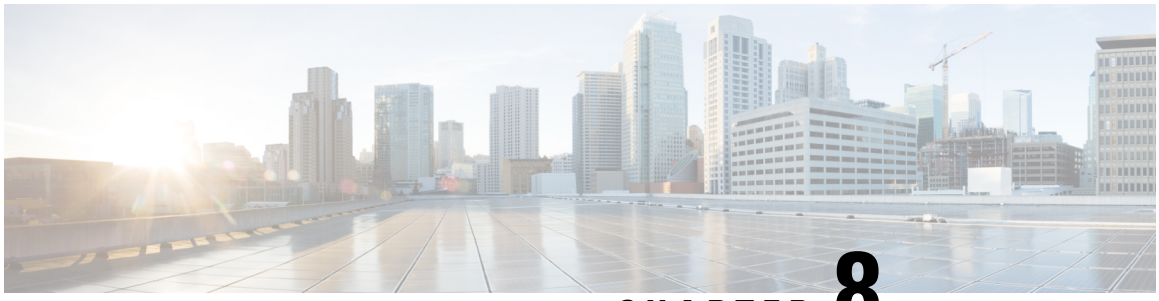
Feature History for Per-Port MTU

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

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

Release	Feature	Feature Information
Cisco IOS XE Amsterdam 17.1.1	Per-Port MTU	Per-Port MTU defines the maximum transmission unit size for frames received and transmitted on a particular port or port channel.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 8

Configuring Internal Power Supplies

- Information About Internal Power Supplies , on page 85
- How to Configure Internal Power Supplies, on page 85
- Monitoring Internal Power Supplies, on page 86
- Configuration Examples for Internal Power Supplies, on page 86
- Additional References for Internal Power Supplies, on page 87
- Feature History for Internal Power Supplies, on page 87

Information About Internal Power Supplies

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

How to Configure Internal Power Supplies

Configuring Internal Power Supply

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

Follow these steps beginning in user EXEC mode:

Procedure

	Command or Action	Purpose
Step 1	**power supply** *switch_number* **slot**{**A** | **B**} { **off** | **on** } **Example:** `Device# power supply 1 slot A on`	Sets the specified power supply to **off** or **on** by using one of these keywords: • **A** —Selects the power supply in slot A. • **B** —Selects power supply in slot B. **Note** Power supply slot B is the closest to the outer edge of the device. • **off** —Set the power supply off.

	Command or Action	Purpose
		• **on** —Set the power supply on. By default, the device power supply is **on**.
Step 2	**show environment power** **Example:** `Device# show environment power`	Verifies your settings.

Monitoring Internal Power Supplies

Table 11: Show Commands for Power Supplies

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

Configuration Examples for Internal Power Supplies

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

```
Device# power supply 1 slot A off
Disabling Power supply A may result in a power loss to PoE devices and/or switches ...
Continue? (yes/[no]): yes
Device#
Jun 10 04:52:54.389: %PLATFORM_ENV-6-FRU_PS_OIR: FRU Power Supply 1 powered off
Jun 10 04:52:56.717: %PLATFORM_ENV-1-FAN_NOT_PRESENT: Fan is not present
Device#
```

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

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

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

Table 12: show env power Status Descriptions

Field	Description
OK	The power supply is present and power is good.

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

Additional References for Internal Power Supplies

Related Documentation

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	*Command Reference (Catalyst 9300 Series Switches)*
For information about the power supplies.	*Cisco Catalyst 9300 Series Switches Hardware Installation Guide*

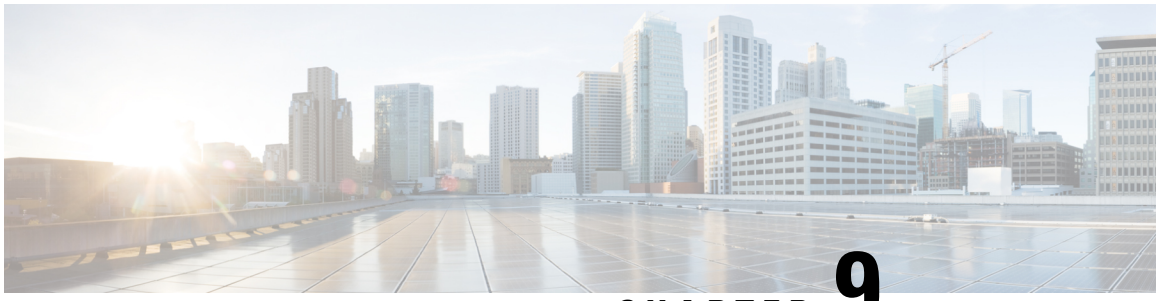
Feature History for Internal Power Supplies

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

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

Release	Feature	Feature Information
Cisco IOS XE Everest 16.5.1a	Internal Power Supplies	The switch operates with power supply modules which could be AC, DC or both. Refer the *Hardware Installation Guide* for more details on power supply units.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 9

Configuring the Cisco Expandable Power System 2200

This module contains the following sections:

- Restrictions for Configuring the Expandable Power System 2200, on page 89
- Information About Cisco Expandable Power System 2200, on page 89
- How to Configure the Cisco Expandable Power System 2200, on page 93
- Monitoring and Maintaining the Cisco Expandable Power System 2200, on page 96
- Additional References for Cisco Expandable Power System 2200, on page 96
- Feature History for Cisco Expandable Power System 2200, on page 97

Restrictions for Configuring the Expandable Power System 2200

- When using the Expandable Power System (XPS) power supplies in the RPS mode for backing up switch power supplies, the smallest power supply in the XPS must be greater than the largest power supply in a switch connected to an XPS port in RPS mode.
- In RPS mode, each XPS power supply can back up only one switch power supply, regardless of the size.
- If you remove a power supply from the power stack (from a switch or the XPS), be sure that removing it does not deplete available power enough to cause load shedding.
- Cisco Catalyst 9300L Series Switches do not support Stack Power and XPS 2200.
- Cisco Catalyst 9300X Series Switches do not support XPS 2200.

Information About Cisco Expandable Power System 2200

The following sections provide an overview of XPS 2200 and its Power Supply Modes.

Cisco eXpandable Power System (XPS) 2200 Overview

The Cisco eXpandable Power System (XPS) 2200 is a standalone power system that you can connect to Catalyst switches. The XPS 2200 can provide backup power to connected devices that experience a power supply failure or, in a Catalyst switch power stack, it can supply additional power to the power stack budget.

The XPS 2200 power ports and internal power supplies can operate in redundant power supply (RPS) mode or stack power (SP) mode.

Stack-power mode is used only on stacking-capable switches in a power stack. With no XPS, a power stack operates in ring topology with a maximum of four switches in the stack. If you merge two stacks, the total number of switches cannot exceed four. When an XPS is in the power stack, you can connect up to nine switches in the stack plus the XPS, providing power budgets to power stack members similar to stack-power ring topology operation.

All Catalyst switches connected to an XPS on SP ports are part of the same power stack, and all power from the XPS and the switch is shared across all switches in the stack. Power sharing is the default mode, but the XPS supports the same stack power modes that are supported in a ring topology (strict and nonstrict power-sharing or redundant modes).

When two power supplies are present, the system can operate in mixed mode, where one power supply operates in RPS mode and the other in SP mode. You can configure the ports and power supplies for the way that you plan to use the XPS 2200.

The XPS 2200 has nine power ports that can operate in an RPS role or in an automatic stack power (Auto-SP) role (the default), where mode of operation is determined by the type of switch connected to the port . You can also use the CLI to force the mode to be RPS for stackable switches.

- When a Catalyst (stackable) switch running the Network Essentials or Network Advantage license is connected to the port, the mode is SP, which enables the switch to be part of the stack power system.

You configure the XPS through any switch connected to a power port. You can use any XPS port for configuration, and you can configure any port from any switch connected to the XPS. If you enter XPS configuration commands on more than one switch, the last configuration applied takes effect.

Although all XPS configuration is done through a switch, the XPS 2200 also runs its own software. You can upgrade this software through the XPS Service Port.

XPS 2200 Power Supply Modes

The XPS has two power supplies that can also be in either RPS or SP mode.

In SP mode, all SP ports on the XPS belong to the same power stack. When a power stack includes an XPS, the stack topology is a star topology and consists of up to nine member switches plus the XPS 2200. The XPS power supply or power supplies that are in SP mode are considered in the power budgeting. If both XPS power supplies are in RPS mode, the power stack consists only of the switches connected to XPS ports in SP mode, and the power budget is determined by the power supplies in these switches.

If there is a power supply role mismatch, for example, if an XPS port is configured for RPS and both power supplies are in SP mode, the XPS detects the mismatch, and an error message is sent.

RPS Mode

When both XPS power supplies are in RPS mode, the XPS can back up two power supply failures for switch power supplies of equal value or less. The smallest power supply in the XPS must be greater than the largest power supply in a switch connected to an XPS port in RPS mode.

If only one supply is in RPS mode, the XPS can back up only one power supply, even when the failed power supply is much smaller. For example, if an XPS 1100 W power supply is in RPS mode and two 350 W switch power supplies fail, the XPS can back up only one of the switch power supplies.

When one XPS power supply in RPS mode is backing up a switch power supply and another switch power supply fails, a message appears that the XPS backup is not available. When the failed power supply comes up, the XPS becomes available to back up other power supplies.

If the XPS is backing up two failed power supplies in a single switch (both XPS power supplies in RPS mode), the XPS is not available to back up other switch power supplies until both of the failed supplies are repaired or replaced.

In mixed mode, with one power supply in RPS mode and one in SP mode, if two power supplies in a single switch fail, because the XPS can back up only one of them, it denies power to both power supplies, and the switch shuts down. This occurs only in mixed power mode.

If a switch is connected to a port configured as RPS, but neither of the power supplies is RPS, the RPS port configuration is rejected and the XPS attempts to add the switch to a power stack. If the switch is not capable of operating in SP mode (is not a stackable switch), the port is disabled.

Ports in RPS mode have a configurable priority. The default priority is based on the XPS port number, with port 1 as the highest priority port. A higher priority port has a higher precedence for backup than a lower priority port. If a switch connected to a higher priority port has a power supply failure while a switch connected to a low priority port is being backed up, the XPS drops power to the low priority port to supply power to the high priority port.

Stack Power Mode

Stack-power mode is used only on Catalyst switches in a power stack. With no XPS, a power stack operates in ring topology with a maximum of four switches in the stack. When an XPS is in the power stack, you can connect up to nine switches in the stack plus the XPS, providing power budgets to power stack members similar to stack-power ring topology operation.

All Catalyst switches connected to an XPS on SP ports are part of the same power stack, and all power from the XPS and the switches is shared across all switches in the stack. Power sharing is the default mode, but the XPS supports the same stack power modes that are supported in a ring topology (strict and nonstrict power-sharing or redundant modes).

The XPS uses neighbor discovery to create the power stack. When it discovers a Catalyst switch on an unconfigured port, it marks the port as an SP port, and the switch joins the power stack. The XPS notifies the switch, begins the power-budgeting process, and assigns budgets to each switch in the power stack based on their requirements, priorities, current power allocations, and the stack aggregate power capability.

The XPS sends the power budget to each switch. If not enough input power is available to provide every switch with its maximum requested power, power is distributed based on priority. Switches with the highest priority receive required power first, followed by any powered devices that have already been allocated power, in order of their priority. Any remaining power is distributed equally through the stack.

The RPS port priority (1 through 9) does not affect stack power priority. Each switch participating in stack power has its own system priority and a high and low priority for devices connected to its ports. These priorities are used for stack power, as is the case in a ring topology. You configure stack power priority for the system and for high and low-priority ports by using the **power-priority switch**, **power-priority high**, and **power-priority low** commands in switch stack power configuration mode. If a system or set of powered devices are using the default priority, the XPS automatically assigns a priority (1 through 27), with lower MAC addresses receiving higher priorities.

There are four power stack modes: power sharing, strict power sharing, redundant, or strict redundant. You configure the power stack mode by using the **mode** {**power-sharing** | **redundant**} [**strict**] command in power-stack configuration mode. The **power-sharing** or **redundant** configurations affect the power budgeting

aspect of the stack; **strict** or non-strict affects the actions of the PoE application when a budget reduction does not result in load shedding.

- In power sharing modes (strict or nonstrict), the stack power budget is the cumulative capacity of all the power supplies in the stack (minus 30 W reserved power). This is the default.
- In redundant modes (strict or nonstrict), the stack power budget is the total available power (minus 30 W) after the capacity of the largest power supply in the power stack is subtracted. Redundant mode guarantees that no switch or powered device loses power or experiences load sheds if a single power supply fails, but load sheds can occur if more than one power supply fails.
- In strict modes, if a loss of input power results in reduced power budgets but does not result in any hardware load shedding, the XPS automatically begins denying power to low-priority powered devices and then the high-priority powered devices until the amount of allocated power is less than or equal to the amount of available PoE power.
- In nonstrict modes, in the event of a power reduction, the amount of allocated power is allowed to fall under budget.

For example, a system with a total PoE budget (available power) of 400 W can allocate 390 W of the budget (allocated power) to powered devices. The allocated power of a device is the maximum amount of power that the device needs. The actual power consumption (consumed power) for a set of powered devices is usually not equal to the allocated power. In this example, the actual power might be approximately 200 W. If a power loss in the stack reduces the available power to 210 W, this amount is enough to sustain the power being consumed by the powered devices, but less than the worst-case allocated power, which would put the system *under budget*. In strict mode, the stack would immediately deny power to powered devices until the allocated power was 210 W or less. In nonstrict mode, no action is taken, and the state is allowed to persist. In nonstrict mode if the actual power consumption becomes more than 210 W, this triggers a load shed and can result in the loss of power to all powered devices or switches with the lowest priority level.

Mixed Modes

The XPS 2200 can also operate in mixed mode, where some ports connected to switches are RPS and others are SP. At least one power supply must be an RPS power supply in this configuration. The power supply in the XPS can back up only one switch power supply and the XPS supply must be greater than the largest power supply in a switch connected to an XPS port in RPS mode.

Switches connected to SP ports belong to a single power stack. If the SP switches have a large enough power budget, an SP power supply is not required on the XPS. When an XPS power supply is configured, its power is added to the power pool shared by the power stack.

XPS 2200 System Defaults

The default role for a port is Auto-SP, where the power mode is determined by the switch connected to the port (SP for Catalyst switches with the Network Essentials or Network Advantage license)

The default for the XPS power supply A (PS1) is RPS mode. The default for power supply B (PS2) is SP mode.

The default mode for all ports and power supplies is enabled.

On ports configured for RPS, the default priority is the same as the port number.

How to Configure the Cisco Expandable Power System 2200

You can configure the XPS from any switch connected to an XPS port. If you enter XPS configuration commands on more than one switch, the last configuration applied takes effect. Only the switch and port name are saved in the switch configuration file.

Configuring System Names

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**power xps** *switch-number* **name** {*name* | **serialnumber**}	**Note** In a stacked system, the switch-number entered must be the switch number of the active switch. Configures a name for the XPS 2200 system. • *name*—Enter a name for the XPS 2200 system. The name can have up to 20 characters. • **serialnumber**—Use the serial number of the XPS 2200 as the system name.
Step 4	**power xps** *switch-number* **port** {*name* | **hostname** | **serialnumber**}	**Note** The *switch-number* appears only on Catalyst switches and represents the device number in the data stack, Configures a name for an XPS 2200 port connected to the device. • *name*—Enter a name for the XPS 2200 port. • **serialnumber**—Use the serial number of the device connected to the port. • **hostname**—Use the hostname of the device connected to the port.
Step 5	**end**	Returns to privileged EXEC mode.
Step 6	**show env xps system**	Verifies the configured name of the system and ports.

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

Configuring XPS Ports

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**power xps** *switch-number* **port** {*number* | **connected**} **mode** {**disable** | **enable**}	**Note** The *switch-number* appears only on Catalyst switches and represents the switch number in the data stack, a value from 1 to 9. Sets the port to be enabled or disabled. • *number*—Enter the XPS 2200 port number. The range is 1 to 9. • **connected**—Enter this keyword if you do not know the port number to which the switch is connected. • **mode disable**—Disable (shut down) the XPS port. **Note** Disabling an XPS port is like removing the cable and appears the same in the **show** command outputs. If the physical cable is connected, you can still use the **enable** keyword to enable the port. • **mode enable**—Enable the XPS port. This is the default.
Step 3	**power xps** *switch-number* **port** {*number* | **connected**} **role** {**auto** | **rps**}	**Note** The *switch-number* represents the switch number in the data stack, a value from 1 to 9. Sets the role of the XPS port. • **role auto**—The port mode is determined by the switch connected to the port. This is the default.

	Command or Action	Purpose
		• **role RPS**—The XPS acts as a back up if the switch power supply fails. At least one RPS power supply must be in RPS mode for this configuration.
Step 4	**power xps** *switch-number* **port** {*number* | **connected**} **priority** *port-priority*	**Note** The *switch-number* represents the switch number in the data stack, a value from 1 to 9. Sets the RPS priority of the port, where higher priority ports take precedence over low priority ports if multiple power supplies fail. This command takes effect only when the port mode is RPS. When the port mode is stack power, you set priority by using the stack power commands. • **priority** *port-priority*—Set the RPS priority of the port. The range is 1 to 9, with 1 being the highest priority. The default priority is the XPS port number.
Step 5	**show env xps port**	Verifies the XPS configuration of the port.
Step 6	**copy running-config startup-config**	(Optional) Saves your entries in the configuration file.

Configuring XPS Power Supplies

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**power xps** *switch-number* **supply** {**A** | **B**} **mode** {**rps** | **sp**}	**Note** The *switch-number* represents the switch number in the data stack, a value from 1 to 9. Sets the XPS power supply mode. • **supply** {**A** | **B**}—Select the power supply to configure. Power supply A is on the left (labeled PS1) and power supply B (PS2) is on the right. • **mode rps**—Set the power supply mode to RPS, to back up connected switches. This

	Command or Action	Purpose
		is the default setting for power supply A (PS1). • **mode sp**—Set the power supply mode to stack power (SP), to participate in the power stack. This is the default setting for power supply B (PS2).
Step 3	**power xps** *switch-number* **supply** {**A** | **B**} {**on** | **off**}	**Note** The *switch-number* represents the switch number in the data stack, a value from 1 to 9. Sets the XPS power supply to be on or off. The default is for both power supplies to be on.
Step 4	**end**	Returns to privileged EXEC mode.
Step 5	**show env xps power**	Displays the status of the XPS power supplies.

Monitoring and Maintaining the Cisco Expandable Power System 2200

Command	Purpose
show env xps system	Verifies the configured name of the system and ports.
show env xps port	Verifies the XPS configuration of the port.
show env xps power	Displays the status of the XPS power supplies.

Additional References for Cisco Expandable Power System 2200

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the *Interface and Hardware Commands* section of the *Command Reference (Catalyst 9300 Series Switches)* .

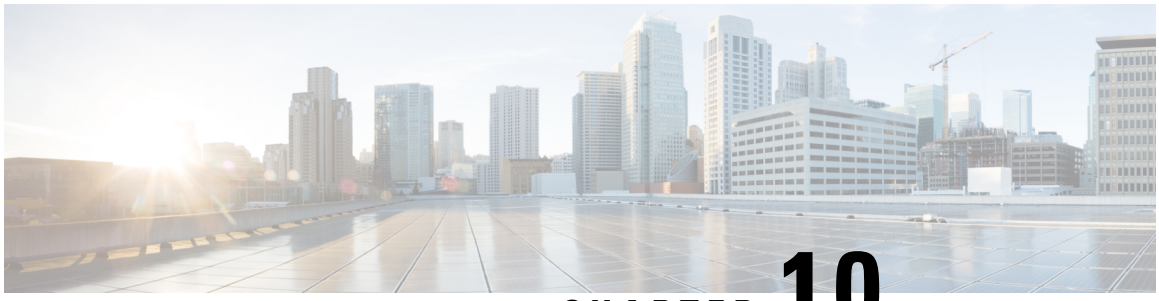
Feature History for Cisco Expandable Power System 2200

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

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

Release	Feature	Feature Information
Cisco IOS XE Everest 16.5.1a	Cisco Expandable Power System (XPS) 2200	The XPS 2200 is a standalone power system that can provide backup power to connected devices that experience a power supply failure; or, in a Catalyst switch power stack, it can supply additional power to the power stack budget. Support for this feature was introduced only on the 9300 switch models of the Cisco Catalyst 9300 Series Switches.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 10

Configuring EEE

- Restrictions for EEE, on page 99
- Information About EEE, on page 99
- How to Configure EEE, on page 100
- Monitoring EEE, on page 101
- Configuration Examples for Configuring EEE, on page 102
- Additional References for EEE, on page 102
- Feature History for Configuring EEE, on page 102

Restrictions for EEE

Energy Efficient Ethernet (EEE) has the following restrictions:

- Changing the EEE configuration resets the interface because the device has to restart Layer 1 autonegotiation.
- You might want to enable the Link Layer Discovery Protocol (LLDP) for devices that require longer wakeup times before they are able to accept data on their receive paths. Doing so enables the device to negotiate for extended system wakeup times from the transmitting link partner.
- If a Multigigabit Ethernet port link is negotiated to 100 Mbps speeds, EEE will not initiate power-saving on the device.
- EEE is not supported on the following switches:
  - C9300-24S
  - C9300-48S

Information About EEE

EEE Overview

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

Default EEE Configuration

EEE is disabled by default.

How to Configure EEE

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

Enabling or Disabling EEE

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 1/0/1`	Specifies the interface to be configured, and enter interface configuration mode.
Step 3	**power efficient-ethernet auto** **Example:** `Device(config-if)# power efficient-ethernet auto`	Enables EEE on the specified interface. When EEE is enabled, the device advertises and autonegotiates EEE to its link partner.
Step 4	**no power efficient-ethernet auto** **Example:** `Device(config-if)# no power efficient-ethernet auto`	Disables EEE on the specified interface.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

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

Monitoring EEE

Table 13: Commands for Displaying EEE Settings

Command	Purpose
show eee capabilities interface *interface-id*	Displays EEE capabilities for the specified interface.
show eee status interface *interface-id*	Displays EEE status information for the specified interface.
show eee counters interface *interface-id*	Displays EEE counters for the specified interface. **Note** Starting from Cisco IOS XE Gibraltar 16.12.1, th **counters interface** *interface-id* command is not s switch models with Multigigabit (mGig) Ethernet

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

```
Switch#show eee capabilities interface gigabitEthernet2/0/1
Gi2/0/1
EEE(efficient-ethernet): yes (100-Tx and 1000T auto)
Link Partner : yes (100-Tx and 1000T auto)

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

```
Switch#show eee status interface gigabitEthernet2/0/1
Gi2/0/1 is up
EEE(efficient-ethernet): Operational
Rx LPI Status : Low Power
Tx LPI Status : Low Power
Wake Error Count : 0

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

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

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

Configuration Examples for Configuring EEE

This example shows how to enable EEE for an interface:

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

This example shows how to disable EEE for an interface:

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

Additional References for EEE

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the *Interface and Hardware Commands* section of the *Command Reference (Catalyst 9300 Series Switches)* .

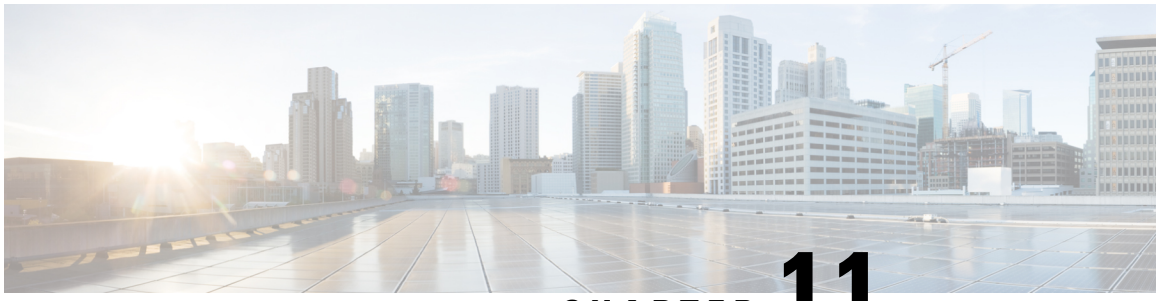
Feature History for Configuring EEE

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

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

Release	Feature	Feature Information
Cisco IOS XE Everest 16.5.1a	Energy Efficient Ethernet	Energy Efficient Ethernet (EEE) is an IEEE 802.3az standard that is designed to reduce power consumption in Ethernet networks during idle periods.
Cisco IOS XE Gibraltar 16.12.1	EEE on Multigigabit (mGig) Ethernet ports	Energy Efficient Ethernet was introduced on switch models with mGig Ethernet ports.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 11

Configuring Power over Ethernet

- Information About Power over Ethernet, on page 103
- How to Configure PoE and UPOE, on page 109
- Monitoring Power Status, on page 114
- Additional References for Power over Ethernet, on page 118
- Feature History for Power over Ethernet, on page 118

Information About Power over Ethernet

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

PoE and PoE+ Ports

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

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

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

Note The following Cisco Catalyst 9300 Switches do not support PoE:

- C9300-24T
- C9300-48T
- C9300-24S
- C9300-48S
- C9300L-24T
- C9300L-48T

Supported Protocols and Standards

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

- **Cisco Discovery Protocol (CDP) with power consumption:** The powered device notifies the device of the amount of power it is consuming. The device does not reply to the power-consumption messages. The device can only supply power to or remove power from the PoE port.
- **Cisco intelligent power management:** The powered device and the device negotiate through power-negotiation CDP messages for an agreed-upon power-consumption level. The negotiation allows a high-power 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 device.

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

  Cisco intelligent power management is backward-compatible with CDP with power consumption; the device responds according to the CDP message that it receives. CDP is not supported on third party-powered devices. Therefore, the device uses the IEEE classification to determine the power usage of the device.
- **IEEE 802.3af:** The major features of this standard are powered-device discovery, power administration, disconnect detection, and optional powered-device power classification.
- **IEEE 802.3at:** The PoE+ standard increases the maximum power that can be drawn by a powered device from 15.4 W per port to 30 W per port.
- The **Cisco Universal Power Over Ethernet (UPOE)** feature provides the capability to source up to 60 W of power (2 x 30 W) over both signal and spare pairs of the RJ-45 Ethernet cable by using the Layer 2 power negotiation protocols such as CDP or LLDP. An LLDP and CDP request of 30 W and higher in the presence of the 4-wire Cisco proprietary spare-pair power type, length, and value descriptions (TLV) can provide power to the spare pair.

  When enabled in IEEE 802.3bt mode, Cisco UPOE devices function as 802.3bt Type 3 devices, supporting up to Class 6 (see Table 14: IEEE Power Classifications in the document) on every port.

Note Only the following UPOE switches are IEEE 802.3bt-complaint Type 3 devices:

- C9300-24U
- C9300-48U
- C9300-24UX
- C9300-48UXM
- C9300-48UN
- C9300L-24UXG-4X
- C9300L-24UXG-2Q
- C9300L-48UXG-4X
- C9300L-48UXG-2Q

Powered-Device Detection and Initial Power Allocation

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

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

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

Table 14: IEEE Power Classifications

Class	Maximum Power Level Required from the Device
0 (class status unknown)	15.4 W
1	4 W
2	7 W
3	15.4 W
4	30 W
5	45 W

Class	Maximum Power Level Required from the Device
6	60 W
7	75 W
8	90 W

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

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

With PoE+, powered devices use IEEE 802.3at and LLDP power with medium-dependent interface (MDI) type, length, and value descriptions (TLVs) and power-via-MDI TLVs, for negotiating power up to 30 W. Cisco prestandard devices and Cisco IEEE powered devices can use CDP or the IEEE 802.3 at power-via-MDI power-negotiation mechanism to request power levels up to 30 W.

Note The 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 Cisco Catalyst Switches software configuration guides and command references.

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

The PoE feature operates the same whether or not the device is a stack member. The power budget is per device and independent of any other devices in the stack. Election of a new active device does not affect PoE operation. The active device keeps track of the PoE status for all devices and ports in the stack and includes the status in output displays.

The stacking-capable device also supports StackPower, which allows the power supplies to share the load across multiple systems in a stack when you connect the device with power stack cables. You can manage the power supplies of up to four stack members as one large power supply.

Note Cisco Catalyst 9300L Series Switches do not support StackPower.

Power Management Modes

The device supports these PoE modes:

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

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

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

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

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

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

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

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

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

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

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

Stacking-capable device also support StackPower, which allows device power supplies to share the load across multiple systems in a stack by connecting up to four device with power stack cables.

Power Monitoring and Power Policing

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

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

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

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

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

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

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

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

Power Consumption Values

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

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

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

device does not police the real-time power consumption of the device, and the device can consume more power than the maximum allocated amount, which could adversely affect the device and the devices connected to the other PoE ports.

Because a standalone device supports internal power supplies, the total amount of power available for the powered devices varies depending on the power supply configuration.

- If a power supply is removed and replaced by a new power supply with less power and the device does not have enough power for the powered devices, the device denies power to the PoE ports in auto mode in descending order of the port numbers. If the device still does not have enough power, the device then denies power to the PoE ports in static mode in descending order of the port numbers.
- If the new power supply supports more power than the previous one and the device now has more power available, the device grants power to the PoE ports in static mode in ascending order of the port numbers. If it still has power available, the device then grants power to the PoE ports in auto mode in ascending order of the port numbers.

The stacking-capable device also supports StackPower, which allows power supplies to share the load across multiple systems in a stack by connecting the device with power stack cables. You can collectively manage the power supplies of up to four stack members as one large power supply.

Cisco Universal Power Over Ethernet

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

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

A device can support Cisco UPOE and 802.3bt Type 3 on the same port. Note that while an 802.3bt-compliant Type 3 device and a Cisco UPOE device offer 60 W, they operate differently. 802.3bt-compliant devices mutually identify the maximum power requirements during physical classification (see Table 14: IEEE Power Classifications). An 802.3bt-compliant Type 3 powered device cannot ask for more power over LLDP than what is requested over the physical layer, which means that an 802.3bt-compliant Class 4 powered device cannot ask for more than 30 W using CDP or LLDP. However, an 802.3bt-compliant Class 6 powered device requests 60 W from the physical layer immediately before the data link layer is established.

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

How to Configure PoE and UPOE

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

Configuring a Power Management Mode on a PoE Port

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet2/0/1`	Specifies the physical port to be configured, and enters interface configuration mode.
Step 4	**power inline** {**auto** [**max** *max-wattage*] | **never** | **static** [**max** *max-wattage*] } **Example:** `Device(config-if)# power inline auto`	Configures the PoE mode on the port. The following are the keywords: • **auto**: Enables detection of powered devices. If enough power is available, automatically allocates power to the PoE port after device detection. This is the default setting. • **max** *max-wattage*: Limits the power allowed on the port. The range for Cisco UPOE ports is 4000 to 60000 milliWatts (mW). If no value is specified, the maximum is allowed. • **never**: Disables device detection and power to the port.

	Command or Action	Purpose
		Note If a port has a Cisco-powered device connected to it, do not use the **power inline never** command to configure the port. A false link-up can occur, placing the port in the error-disabled state. • **static**: Enables detection of powered devices. Preallocate (reserve) power for a port before the device discovers the powered device. The device reserves power for this port even when no device is connected, and guarantees that power will be provided upon device detection. 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 the PoE status for a device or a device stack, for the specified interface, or for a specified stack member. The **module** *switch-number* keywords are supported only on stacking-capable devices.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Enabling Power on Signal and Spare Pairs

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

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet2/0/1`	Specifies the physical port to be configured, and enters interface configuration mode.
Step 3	**power inline four-pair forced** **Example:** `Device(config-if)# power inline four-pair forced`	(Optional) Enables power on both signal and spare pairs from a switch port.
Step 4	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuring Power Policing

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

Procedure

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

	Command or Action	Purpose
Step 4	**power inline police** [**action** {**log** | **errdisable**}] **Example:** `Device(config-if)# power inline police`	Configures the device to take one of these actions if the real-time power consumption exceeds the maximum power allocation on the port: • **power inline police**: Shuts down the PoE port, turns off power to it, and puts it in the error-disabled state. **Note** You can enable error detection for the PoE error-disabled cause by using the **errdisable detect cause inline-power** global configuration command. You can also enable the timer to recover from the PoE error-disabled state by using the **errdisable recovery cause inline-power interval** *interval* global configuration command. • **power inline police action errdisable**: Turns off power to the port if the real-time power consumption exceeds the maximum power allocation on the port. • **power inline police action log**: Generates a syslog message while still providing power to the port. If you do not enter the **action log** keywords, the default action shuts down the port and puts the port in the error-disabled state.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.

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

	Command or Action	Purpose
	`Device(config)# errdisable recovery interval 100`	
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 verifies the error recovery settings.

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

Enable the 802.3bt Mode on Type 3 UPOE Modules

The Cisco Catalyst 9300 UPOE switches that support IEEE 802.3bt standard for Type 3 powered devices are in 802.3at mode, by default. You can enable 802.3bt mode on them using the **hw-module switch** *switch_no* **upoe-plus** command in the global configuration mode. Note that the **hw-module switch** *switch_no* **upoe-plus** command power cycles the switch upon configuration.

```
Device(config)# hw-module switch 2 upoe-plus
!!!WARNING!!!This configuration will power cycle the switch to make it effective. Would you
 like to continue y/n?
Device#y
```

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

Monitoring Power Status

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

Table 15: show Commands for Power Status

Command	Purpose
show env power switch [*switch-number*]	(Optional) Displays the status of the internal power supplies for each switch in the stack or for the specified switch. The range is 1 to 9, depending on the switch member numbers in the stack. These keywords are available only on stacking-capable switches.
show power inline [*interface-id* | **module** *switch-number*]	Displays PoE status for a switch or switch stack, for an interface, or for a specific switch in the stack.
show power inline police	Displays power-policing data.
show power inline upoe-plus [*interface-id*] [**module**]	Displays the PoE status for an interface that is enabled for 802.3bt-compliant mode.

Examples

```
Device# show power inline upoe-plus

Module   Available     Used     Remaining
          (Watts)     (Watts)    (Watts)
------   ---------   --------   ---------
3          1310.0      660.0       650.0
Codes: DS - Dual Signature device, SS - Single Signature device
       SP - Single Pairset device

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

Te3/0/1     auto   DS   on,on         60.0      6.8       3,3     Ieee PD
Te3/0/2     auto   DS   on,on         60.0      6.7       3,3     Ieee PD
Te3/0/3     auto   DS   on,on         60.0      6.8       3,3     Ieee PD
Te3/0/4     auto   DS   on,on         60.0      6.8       3,3     Ieee PD
Te3/0/5     auto   DS   on,on         60.0      6.8       3,3     Ieee PD
Te3/0/6     auto   DS   on,on         60.0      6.8       3,3     Ieee PD
Codes: DS - Dual Signature device, SS - Single Signature device
       SP - Single Pairset device

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

Te3/0/7     auto   DS   on,on         60.0      6.8       3,3     Ieee PD
Te3/0/8     auto   DS   on,on         60.0      6.8       3,3     Ieee PD
Te3/0/9     auto   n/a  off           0.0       0.0       n/a
Te3/0/10    auto   SS   on,off        30.0      5.4       4       Ieee PD
Te3/0/11    auto   SS   on,off        30.0      9.0       4       Ieee PD
Te3/0/12    auto   SS   on,off        30.0      9.7       4       Ieee PD
Te3/0/13    auto   n/a  off           0.0       0.0       n/a
Te3/0/14    auto   n/a  off           0.0       0.0       n/a
Te3/0/15    auto   n/a  off           0.0       0.0       n/a
Te3/0/16    auto   n/a  off           0.0       0.0       n/a
Te3/0/17    auto   n/a  off           0.0       0.0       n/a
Codes: DS - Dual Signature device, SS - Single Signature device
       SP - Single Pairset device
```

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

Te3/0/18    auto   n/a  off           0.0       0.0       n/a
Te3/0/19    auto   n/a  off           0.0       0.0       n/a
Te3/0/20    auto   n/a  off           0.0       0.0       n/a
Te3/0/21    auto   n/a  off           0.0       0.0       n/a
Te3/0/22    auto   SS   on,off        30.0      12.0      4       Ieee PD
Te3/0/23    auto   SS   on,off        30.0      12.3      4       Ieee PD
Te3/0/24    auto   SS   on,off        30.0      5.3       4       Ieee PD
--------- ------ ---------- ---------- ---------- ------------------- -----

Totals:            14   on          660.0    107.9
```

The following are descriptions of the fields that you see in the output of the **show power inline upoe-plus** command:

Table 16: Fields Displayed in the Output of the show power inline upoe-plus Command

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

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

```
Device# show power inline Te3/0/1 detail

 Interface: Te3/0/1
 Inline Power Mode: auto
 Operational status (Alt-A,B): on,on
 Device Detected: yes
 Device Type: Ieee PD
 Connection Check: DS
 IEEE Class (Alt-A,B): 3,3
 Physical Assigned Class (Alt-A,B): 3,3
 Discovery mechanism used/configured: Ieee and Cisco
 Police: off

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

 Actual consumption
 Measured at the port(watts) (Alt-A,B): 3.4,3.3
```

```
 Maximum Power drawn by the device since powered on: 6.9

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

 Power Negotiation Used: None
 LLDP Power Negotiation       --Sent to PD--      --Rcvd from PD--
 Power Type:                  -                    -
 Power Source:                -                    -
 Power Priority:              -                    -
 Requested Power(W):          -                    -
 Allocated Power(W):          -                    -

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

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

```
Device# show power inline police

 Module   Available     Used     Remaining
          (Watts)     (Watts)    (Watts)
------   ---------   --------   ---------
3          1310.0      660.0       650.0
Interface Admin  Oper       Admin      Oper       Cutoff Oper
          State  State      Police     Police     Power  Power
--------- ------ ---------- ---------- ---------- ------ -----
Te3/0/1   auto   on         none       n/a        n/a    6.8
Te3/0/2   auto   on         none       n/a        n/a    6.7
Te3/0/3   auto   on         none       n/a        n/a    6.9
Te3/0/4   auto   on         none       n/a        n/a    6.8
Te3/0/5   auto   on         none       n/a        n/a    6.8
Te3/0/6   auto   on         none       n/a        n/a    6.8
Te3/0/7   auto   on         none       n/a        n/a    6.8
Te3/0/8   auto   on         none       n/a        n/a    6.8
Te3/0/9   auto   off        none       n/a        n/a    n/a
Te3/0/10  auto   on         none       n/a        n/a    5.4
Te3/0/11  auto   on         none       n/a        n/a    8.9
Te3/0/12  auto   on         none       n/a        n/a    9.5
Te3/0/13  auto   off        none       n/a        n/a    n/a
Te3/0/14  auto   off        none       n/a        n/a    n/a
Te3/0/15  auto   off        none       n/a        n/a    n/a
Interface Admin  Oper       Admin      Oper       Cutoff Oper
          State  State      Police     Police     Power  Power
--------- ------ ---------- ---------- ---------- ------ -----
Te3/0/16  auto   off        none       n/a        n/a    n/a
Te3/0/17  auto   off        none       n/a        n/a    n/a
Te3/0/18  auto   off        none       n/a        n/a    n/a
Te3/0/19  auto   off        none       n/a        n/a    n/a
Te3/0/20  auto   off        none       n/a        n/a    n/a
Te3/0/21  auto   off        none       n/a        n/a    n/a
Te3/0/22  auto   on         none       n/a        n/a    12.0
Te3/0/23  auto   on         none       n/a        n/a    12.2
Te3/0/24  auto   on         none       n/a        n/a    5.3
--------- ------ ---------- ---------- ---------- ------ -----
Totals:                                                  107.6
```

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

```
Device# show power inline priority

Interface  Admin  Oper       Admin
           State  State      Priority
---------- ------ ---------- ----------
Te3/0/1    auto   on         low
Te3/0/2    auto   on         low
Te3/0/3    auto   on         low
Te3/0/4    auto   on         low
Te3/0/5    auto   on         low
Te3/0/6    auto   on         low
Te3/0/7    auto   on         low
Te3/0/8    auto   on         low
Te3/0/9    auto   off        low
Te3/0/10   auto   on         high
Te3/0/11   auto   on         high
Te3/0/12   auto   on         high
Te3/0/13   auto   off        high
Te3/0/14   auto   off        high
Te3/0/15   auto   off        high
Te3/0/16   auto   off        high
Te3/0/17   auto   off        low
Te3/0/18   auto   off        low
Te3/0/19   auto   off        low
Te3/0/20   auto   off        high
Te3/0/21   auto   off        high
Te3/0/22   auto   on         high
Te3/0/23   auto   on         high
Te3/0/24   auto   on         high
```

Additional References for Power over Ethernet

Related Documents

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

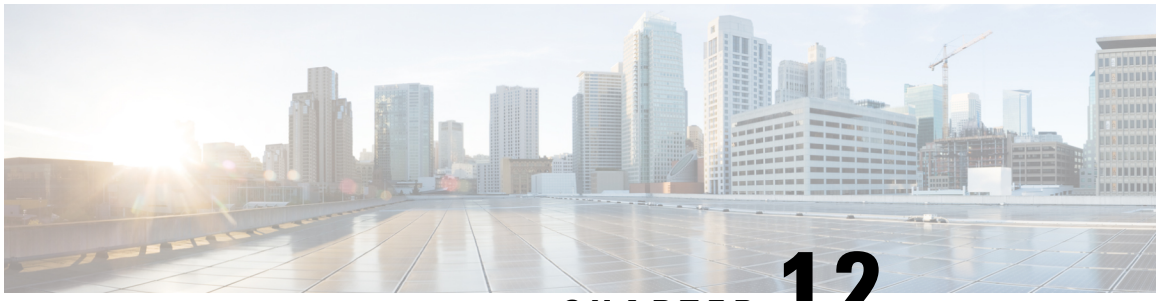
Feature History for Power over Ethernet

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

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

Release	Feature	Feature Information
Cisco IOS XE Everest 16.5.1a	Power over Ethernet (PoE)	Power over Ethernet (PoE) allows the LAN switching infrastructure to provide power to an endpoint, called a powered device, over a copper Ethernet cable. The following types of end points can be powered through PoE: • A Cisco prestandard powered device An IEEE 802.3af-compliant powered device An IEEE 802.3at-compliant powered device
Cisco IOS XE Gibraltar 16.12.1	Support for IEEE 802.3bt Type 3	Support for 802.3bt Type 3 compliance on the Cisco Catalyst 9300 UPOE Series Switches was introduced. The command **hw-module switch upoe-plus** was introduced to enable the 802.3bt mode on the device.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 12

Configuring Perpetual PoE and Fast POE

- Restrictions for Perpetual and Fast PoE, on page 121
- Perpetual POE, on page 122
- Fast POE, on page 122
- Configuring Perpetual and Fast POE, on page 122
- Example: Configuring Perpetual and Fast POE, on page 123
- Feature Information for Persistent and Fast PoE, on page 124

Restrictions for Perpetual and Fast PoE

The following restrictions apply to perpetual and fast PoE :

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

Note Perpetual PoE and Fast PoE are not supported on the following Cisco Catalyst 9300 SKUs:

- C9300-24S
- C9300-48S

Perpetual POE

The Perpetual POE provides uninterrupted power to connected powered device (PD) even when the power sourcing equipment (PSE) switch is reloading and booting up.

Note Power to the ports will be interrupted in case of MCU firmware upgrade and ports will be back up immediately after the upgrade.

Fast POE

This feature remembers the last power drawn from a particular PSE port and switches on power the moment AC power is plugged in (within 15 to 20 seconds of switching on power) without waiting for IOS to boot up. When **poe-ha** is enabled on a particular port, the switch on a recovery after power failure, provides power to the connected endpoint devices within short duration before even the IOS forwarding starts up.

Note In case of UPOE, even though Fast POE is available on the switch side, the PD endpoints may not be able to take advantage of the same, due to the reliance on LLDP to signal the UPOE power availability. This reliance on LLDP requires that the PD endpoint still needs to wait till the IOS comes up and LLDP packet exchanges can happen, signaling the availability of UPOE power.

Configuring Perpetual and Fast POE

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

Note Configure the **perpetual-poe-ha** command before connecting the PD, or manually shut or unshut the port after configuring **poe-ha** command.

Procedure

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

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet 2/0/1`	Specifies the physical port to be configured, and enters interface configuration mode.
Step 4	**power inline port perpetual-poe-ha** **Example:** `Device(config-if)# power inline port perpetual-poe-ha`	Configures perpetual PoE. When you configure perpetual PoE on a port connected to a PD device, the PD device remains powered on during reload.
Step 5	**power inline port poe-ha** **Example:** `Device(config-if)# power inline port poe-ha`	Configures Fast PoE. When you configure Fast PoE, if the switch is power cycled, PD device is powered on within 10-15 seconds of plugging into a power source without waiting for IOS to boot up. **Note** You should configure perpetual PoE using **power inline port perpetual-poe-ha** command before configuring Fast PoE using **power inline port poe-ha** command.
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Example: Configuring Perpetual and Fast POE

This example shows how you can configure perpetual PoE on the switch.

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

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

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

This example shows what happens if you configure fast PoE before configuring perpetual PoE.

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet2/0/1
Device(config-if)# power inline port poe-ha
Interface Gi2/0/1:INFO: Please execute "power inline port perpetual-poe-ha"
 configuration command when "power inline port poe-ha" is configured on
the interface to enable fast poe
Device(config-if)# power inline port perpetual-poe-ha
Device(config-if)# end
```

This example shows what happens when you disable perpetual PoE without disabling fast PoE on the interface.

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet2/0/1
Device(config-if)# power inline port poe-ha
Interface Gi2/0/1:INFO: Please execute "power inline port perpetual-poe-ha"
 configuration command when "power inline port poe-ha" is configured on
the interface to enable fast poe
Device(config-if)# power inline port perpetual-poe-ha
Device(config-if)# no power inline port poe-ha
Device(config-if)# power inline port poe-ha
Device(config-if)# no power inline port perpetual-poe-ha
Interface Gi2/0/1:INFO: Please execute "no power inline port poe-ha"
configuration command, as fast poe has no effect without "power inline
port perpetual-poe-ha" configuration on the interface
Device(config-if)# end
```

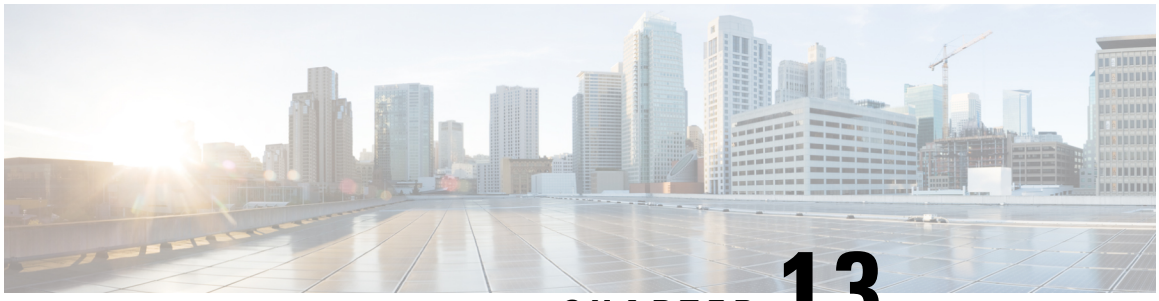
Feature Information for Persistent and Fast PoE

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 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 17: Feature Information for Persistent and Fast PoE

Feature Name	Releases	Feature Information
Perpetual and Fast PoE	Cisco IOS XE Everest 16.5.1a	The Perpetual POE provides uninterrupted power to connected PD device even when the PSE switch is booting. Fast PoE remembers the last power drawn from a particular PSE port and switches on power without waiting for IOS to boot up.



CHAPTER 13

Configuring 2-event Classification

- Restrictions for 2-event classification, on page 125
- Information about 2-event Classification, on page 125
- Configuring 2-event Classification, on page 125
- Example: Configuring 2-Event Classification, on page 126
- Feature History for 2-event Classification, on page 126

Restrictions for 2-event classification

The following restrictions apply to 2-event classification:

- Configuration of 2-event classification has to be done before physically connecting any endpoint. Alternatively do a manual shut/no-shut of the ports drawing power.
- Power to the ports will be interrupted in case of MCU firmware upgrade and ports will be back up immediately after the upgrade.

Information about 2-event Classification

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

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

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

Configuring 2-event Classification

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

Procedure

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

Example: Configuring 2-Event Classification

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

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

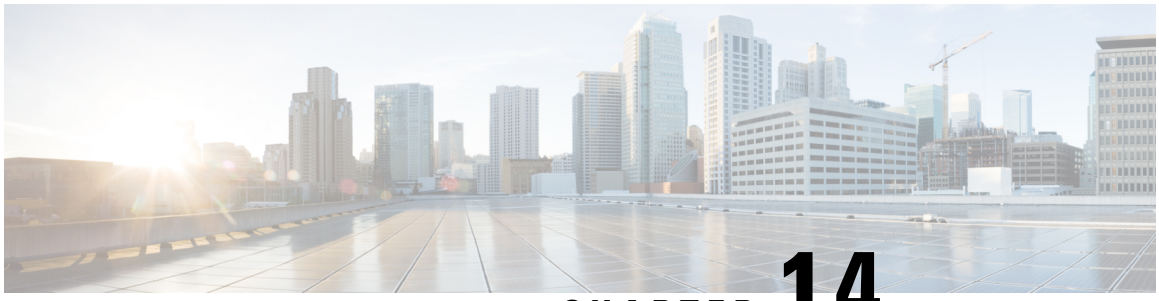
Feature History for 2-event Classification

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

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

Release	Feature	Feature Information
Cisco IOS XE Everest 16.5.1a	2-event classification	When a class 4 device gets detected, IOS allocates 30W without any CDP or LLDP negotiation. This means that even before the link comes up the class 4 power device gets 30W.

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



CHAPTER 14

Configuring Auto SmartPorts

- Restrictions for Auto SmartPorts, on page 129
- Information about Auto SmartPorts, on page 129
- Auto SmartPort Macros, on page 130
- Customizing Device Classifier, on page 130
- Commands executed by CISCO_LIGHT_AUTO_SMARTPORT , on page 130
- Enabling Auto SmartPort, on page 131
- Configuring Device Classifier Profiles, on page 132
- Configuring Mapping Between Event Triggers and Built-in Macros, on page 134
- Example: Enabling Auto SmartPorts, on page 135
- Example: Configuring Mapping Between Event Triggers and Built-in Macros, on page 135
- Example: Configuring Device Classifier Profiles, on page 136
- Feature Information for Auto SmartPorts, on page 136

Restrictions for Auto SmartPorts

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

Information about Auto SmartPorts

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

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

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

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

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

Auto SmartPort Macros

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

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

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

Customizing Device Classifier

The device classifier (DC) collects information from protocols like CDP, LLDP, DHCP and other protocols to identify devices. You must enable CDP and LLDP on the device. To make DHCP options information available to the DC, you must enable the DHCP snooping feature on the switch. The device attributes collected from these protocols are evaluated against a set of profiles available to the DC to find the best match. The best-matched profile is used for device identification.

Device-classifier uses three types of profile definitions—built-in, default, and user-defined profiles.

- Built-in profiles contain the device profiles that are known to the Auto SmartPort module which comprises of a limited set of Cisco devices. They are built into Cisco IOS and cannot be changed.
- Default profiles are stored as a text file in non-volatile storage and allow the DC to identify a much larger set of devices. Default profiles are updated as part of the Cisco IOS archive download.
- User-defined profiles support custom profiling based on user's input. The device classifier identifies rules, conditions and profiles from the user input.

Commands executed by CISCO_LIGHT_AUTO_SMARTPORT

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

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

- switchport mode access
- switchport port-security violation restrict
- switchport port-security mac-address sticky

- switchport port-security
- power inline port poe-ha
- storm-control broadcast level 50.00
- storm-control multicast level 50.00
- storm-control unicast level 50.00
- spanning-tree portfast
- spanning-tree bpduguard enable

Enabling Auto SmartPort

Note Auto SmartPort is disabled by default.

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

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

To enable Auto SmartPorts, perform this task:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**device classifier** **Example:** `Device(config)# device classifier`	Enables the device classifier. Use **no device classifier** command to disable the device classifier.
Step 4	**macro auto global processing** **Example:**	Enables Auto SmartPorts on the switch globally. Use **no macro auto global processing** command to disable Auto SmartPort globally.

	Command or Action	Purpose
	`Device(config)# macro auto global processing`	
Step 5	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 6	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring Device Classifier Profiles

To customise device classifier profiles, follow the steps:

Before you begin

Disable device classifier feature before customising device classifier profiles. Use the **no device classifier** command to disable device classifier feature.

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	**device classifier condition** *condition-name* [**op**{**OR**|**AND**}] **Example:** `Device(config)# device classifier condition ts-cond1`	Defines device classifier condition. • *condition-name* sets the name of the condition for device classifier. • **op OR** defines OR operator of rules. If either of the protocols defined matches, the device gets classified. • **op AND** defines AND operator of rules. If all the protocols defined matches, the device gets classified. **Note** If you change any of the parameters of the condition, it is considered as a new condition. In this case, default **AND** operator of rules is applied for both the conditions.
Step 4	{**cdp**|**dhcp**|**lldp**}**tlv-type** *number* **value**{**integer** *num*|**string** *name*} **Example:** `Device(config-device-classifier-condition)# cdp tlv-type 1 value String TS01`	Configures profiling based on match of TLV for either integer or string value of the given protocol. The protocols supported are CDP, DHCP, and LLDP. • **tlv-type** *number* configures application TLV type information. The *number* range is 1 to 255. • **value** configures application TLV value information. You can set an integer or a string value. **Note** The supported protocols are CDP, DHCP, and LLDP. Protocols like HTTP, OUI are not supported in Cisco IOS XE Bengaluru 17.4.1.
Step 5	**exit** **Example:** `Device(config-device-classifier-condition)# exit`	Exits device classifier condition configuration mode.
Step 6	**device classifier device-type** *profile-name* **Example:** `Device(config)# device classifier device-type Terminal-Server`	Configures profile based on defined conditions. *profile-name* defines a name for the device type.The device gets classified to the set *profile-name* if there is a match of the conditions.

	Command or Action	Purpose
Step 7	**condition** *condition-name* **Example:** `Device(config-device-classifier-dtype)# condition ts-cond1`	Enter name of the condition for the profile.
Step 8	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring Mapping Between Event Triggers and Built-in Macros

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

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

Before you begin

You need to enable auto smartport macros globally.

Procedure

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

	Command or Action	Purpose
Step 4	**macro auto trigger** *event trigger* **Example:** `Device(config)# macro auto trigger CISCO_SWITCH_EVENT`	Invokes the user-defined event trigger.
Step 5	**device** *device_ID* **Example:** `Device(config)# device cisco WS-C3560CX-8PT-S`	Matches the event trigger to the device identifier.
Step 6	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 7	**show shell triggers** **Example:** `Device# show shell triggers`	Displays the event triggers on the switch.
Step 8	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 9	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Example: Enabling Auto SmartPorts

This example shows how you can enable to Auto SmartPort.

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

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

This example shows how you can configure mapping between event triggers and built-in macros.

```
Device> enable
Device# configure terminal
Device(config)# macro auto execute CISCO_SWITCH_EVENT builtin CISCO_SWITCH_AUTO_SMARTPORT
```

```
Device(config)# macro auto trigger CISCO_SWITCH_EVENT
Device(config)# device cisco WS-C3560CX-8PT-S
Device(config)# end
```

Example: Configuring Device Classifier Profiles

The following is a sample configuration of profiling based on match for string in device-name TLV of CDP packet. Once the match is found, the device gets classified to Terminal-Server.

```
Device> enable
Device# configure terminal
Device(config)# device classifier condition ts-cond1
Device(config-device-classifier-condition)# cdp tlv-type 1 value String TS01
Device(config-device-classifier-condition)# exit
Device(config)# device classifier device-type Terminal-Server
Device(config-device-classifier-dtype)#condition ts-cond1
```

The following is a sample configuration of profiling based on match for two different protocols in the device name TLV of CDP packet and system name TLV of LLDP packet. If both the protocol match is found, the device gets classified to Terminal-Server.

```
Device> enable
Device# configure terminal
Device(config)# device classifier condition ts-cond2 op OR
Device(config-device-classifier-condition)# cdp tlv-type 1 value number 0x0029
Device(config-device-classifier-condition)#lldp tlv-type 5 value String TS02
Device(config-device-classifier-condition)# exit
Device(config)#device classifier device-type Terminal-Server
Device(config-device-classifier-dtype)# condition ts-cond2
```

The following is a sample configuration of profiling based on match for two different protocols in the device name TLV of CDP packet and system name TLV of LLDP packet. If both the protocol match is found, the device gets classified to Terminal-Server.

```
Device> enable
Device# configure terminal
Device(config)# device classifier condition ts-cond2 op AND
Device(config-device-classifier-condition)# cdp tlv-type 1 value number 0x0001
Device(config-device-classifier-condition)#lldp tlv-type 5 value String TS02
Device(config-device-classifier-condition)# exit
Device(config)#device classifier device-type Terminal-Server
Device(config-device-classifier-dtype)# condition ts-cond3
```

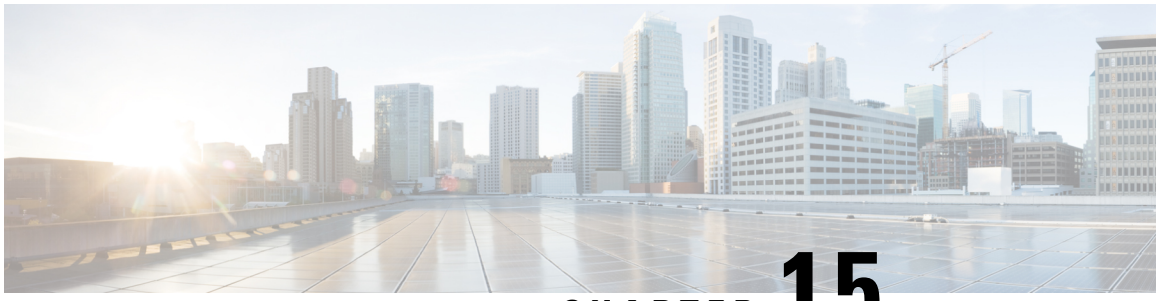
Feature Information for Auto SmartPorts

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 18: Feature Information for Auto SmartPorts

Feature Name	Releases	Feature Information
Auto SmartPorts	Cisco IOS XE Everest 16.5.1a	Auto SmartPort macros dynamically configure ports based on the device type detected on the port. When the switch detects a new device on a port, it applies the appropriate Auto SmartPorts macro.
Device Classifier Profiles	Cisco IOS XE Bengaluru 17.4.1	Allows you to configure the rules for matching and classifying the device using device classifier conditions.



CHAPTER 15

Configuring COAP Proxy Server

- Restrictions for the COAP Proxy Server, on page 139
- Information About the COAP Proxy Server, on page 139
- How to Configure the COAP Proxy Server, on page 140
- Configuration Examples for the COAP Proxy Server, on page 143
- Monitoring COAP Proxy Server, on page 147
- Feature History for COAP, on page 148

Restrictions for the COAP Proxy Server

The following restrictions apply to COAP proxy server:

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

Information About the COAP Proxy Server

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

The comparison of COAP and HTTP is shown below:

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

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

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

How to Configure the COAP Proxy Server

To configure the COAP proxy server, you can configure the COAP Proxy and COAP Endpoints in the Configuration mode.

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

Configuring the COAP Proxy

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**coap proxy** **Example:** `Device(config)# coap proxy`	Enters the COAP proxy sub mode. **Note** To stop the coap proxy and delete all configurations under coap proxy, use the **no coap proxy** command.
Step 4	**security** [**none** [[**ipv4** | **ipv6**] {*ip-address ip-mask/prefix*} | **list** {*ipv4-list name* | *ipv6-list-name*}] | **dtls** [**id-trustpoint** {*identity-trustpoint label*}] [**verification-trustpoint** {*verification-trustpoint*} | [**ipv4** | **ipv6** {*ip-address ip-mask/prefix*}] | **list** {*ipv4-list name* | *ipv6-list-name*}]] **Example:**	Takes the encryption type as argument. The two security modes supported are **none** and **dtls** • none - Indicates no security on that port. With **security none**, a maximum of 5 ipv4 and 5 ipv6 addresses can be associated. • dtls - The DTLS security takes RSA trustpoint and Verification trustpoint

	Command or Action	Purpose
	`Device(config-coap-proxy)# security none ipv4 1.1.0.0 255.255.0.0`	which are optional. Without Verification trustpoint it does the normal Public Key Exchange. With **security dtls**, a maximum of 5 ipv4 and 5 ipv6 addresses can be associated. **Note** To delete all security configurations under coap proxy, use the **no security** command.
Step 5	**max-endpoints** {*number*} **Example:** `Device(config-coap-proxy)#max-endpoints 10`	(Optional) Specifies the maximum number of endpoints that can be learnt on the switch. The default value is 10. The range is 1 to 500. **Note** To delete all max-endpoints configured under coap proxy, use the **no max-endpoints** command.
Step 6	**port-unsecure** {*port-num*} **Example:** `Device(config-coap-proxy)#port-unsecure 5683`	(Optional) Configures a port other than the default 5683. The range is 1 to 65000. **Note** To delete all port configurations under coap proxy, use the **no port-unsecure** command.
Step 7	**port-dtls** {*port-num*} **Example:** `Device(config-coap-proxy)#port-dtls 5864`	(Optional) Configures a port other than the default 5684. **Note** To delete all dtls port configurations under coap proxy, use the **no port-dtls** command.
Step 8	**resource-directory** [ipv4 | ipv6] {*ip-address*}] **Example:** `Device(config-coap-proxy)#resource-directory ipv4 192.168.1.1`	Configures a unicast upstream resource directory server to which the switch can act as a COAP client. With **resource-directory**, a maximum of 5 of ipv4 and 5 ipv6, ip addresses can be configured. **Note** To delete all resource directory configurations under coap proxy, use the **no resource-directory** command.
Step 9	**list** [**ipv4** | **ipv6**] {*list-name*} **Example:** `Device(config-coap-proxy)#list ipv4 trial_list`	(Optional) Restricts the IP address range where the lights and their resources can be learnt. Creates a named list of ip address/masks, to be used in the **security** [**none** | **dtls**] command options above.

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

Configuring COAP Endpoints

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

Procedure

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

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

Step 3	**coap endpoint** [**ipv4** | **ipv6**] {*ip-address*}
	Example:
	`Device(config)#coap endpoint ipv4 1.1.1.1`
	`Device(config)#coap endpoint ipv6 2001::1`
		Configures the static endpoints on the switch.
		• **ipv4** - Configures the IPv4 Static endpoints.
		• **ipv6** - Configures the IPv6 Static endpoints.
		Note To stop the coap proxy on any endpoint, use the **no coap endpoint** [**ipv4** |**ipv6**] {*ip-address*} command.

Step 4	**exit** **Example:** `Device(config-coap-endpoint)# exit`	Exits the COAP endpoint sub mode.
Step 5	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuration Examples for the COAP Proxy Server

Examples: Configuring the COAP Proxy Server

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

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

-----------------------------------------------------------------------------------------
```

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

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

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

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

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

------------------------------------------------------------------------------------------
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
Device#show coap version
CoAP version 1.0.0
RFC 7252
```

```
Device#show coap resources
Link format data =
</>
</1.1.1.6/cisco/context>
</1.1.1.6/cisco/actuator>
</1.1.1.6/cisco/sensor>
</1.1.1.6/cisco/lldp>
</1.1.1.5/cisco/context>
</1.1.1.5/cisco/actuator>
</1.1.1.5/cisco/sensor>
</1.1.1.5/cisco/lldp>
</cisco/flood>
</cisco/context>
</cisco/showtech>
</cisco/lldp>
```

```
Device#show coap globals
Coap System Timer Values :
   Discovery  : 120 sec
   Cache Exp  : 5 sec
   Keep Alive : 120 sec
   Client DB  : 60 sec
   Query Queue: 500 ms
   Ack delay  : 500 ms
   Timeout    : 5 sec

Max Endpoints      : 10
Resource Disc Mode : POST
```

```
Device#show coap stats
Coap Stats :
Endpoints : 2
Requests : 20
Ext Queries : 0
```

```
Device#show coap endpoints
List of all endpoints :

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

```
Endpoints - Total : 2 Discovered : 2 New : 0

-------------------------------------------------------------------------------------------

Device#show coap dtls-endpoints
#     Index State  String State     Value   Port IP
-----------------------------------------------------------------
1     3     SSLOK       3           48969   20.1.1.30
2     2     SSLOK       3           53430   20.1.1.31
3     4     SSLOK       3           54133   20.1.1.32
4     7     SSLOK       3           48236   20.1.1.33


-------------------------------------------------------------------------------------------
```

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

```
Device#debug coap ?
all       Debug CoAP all
database  Debug CoAP Database
errors    Debug CoAP errors
events    Debug CoAP events
packet    Debug CoAP packet
trace     Debug CoAP Trace
warnings  Debug CoAP warnings
```

Monitoring COAP Proxy Server

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

Table 19: Commands to Display to COAP specific data

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

Table 20: Commands to Clear COAP Commands

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

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

Table 21: Commands to Debug COAP protocol

debug coap database	Debugs the COAP database output.

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

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

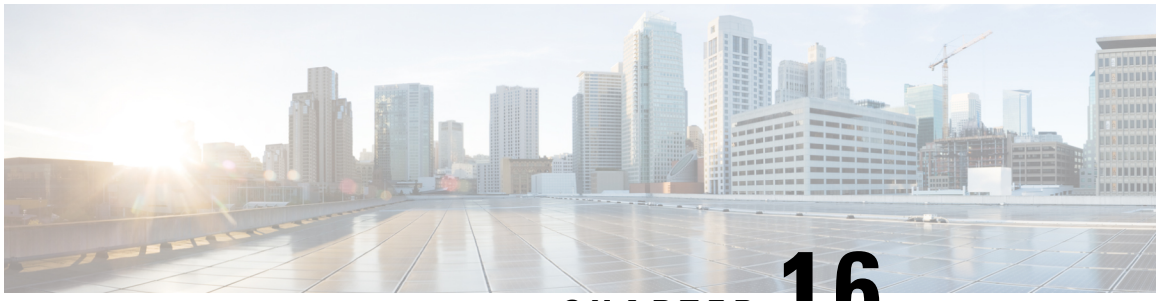
Feature History for COAP

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

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

Release	Feature	Feature Information
Cisco IOS XE Everest 16.5.1a	COAP	The COAP protocol is designed for use with constrained devices. COAP works in the same way on constrained devices as HTTP works on servers in accessing information.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 16

Configuring USB 3.0 SSD

- Information about USB 3.0 SSD, on page 149
- How to Configure USB 3.0 SSD, on page 150
- Monitoring USB 3.0 SSD, on page 153
- Troubleshooting Tips, on page 154
- Configuration Examples for USB 3.0 SSD, on page 157
- Feature History for USB 3.0 SSD, on page 159

Information about USB 3.0 SSD

The following sections provide information about USB 3.0 SSD.

USB 3.0 SSD

In Cisco IOS XE Fuji 16.9.1, support for USB 3.0 SSD is enabled on Cisco Catalyst 9300 Series Switches. USB 3.0 SSD provides extra 240 GB storage for application hosting. Applications can be hosted in Kernel Virtual Machines (KVM), Linux Containers (LXC), or Docker containers. The storage drive can also be used to save packet captures, trace logs generated by the operating system and third-party applications. USB 3.0 SSD can be used simultaneously as a general-purpose storage device and as an application-hosting device. You must use only Cisco USB drives; non-Cisco USB drives are not supported.

Note USB 3.0 SSD cannot be used to boot images, emergency install the images, or upgrade internal flash using (software maintenance update (SMU or **install** commands. Bootloader support for USB 3.0 SSD is not available.

USB 3.0 SSD is enabled with Self-Monitoring, Analysis and Reporting Technology (S.M.A.R.T) functionality for health monitoring of the drive. The purpose of S.M.A.R.T is to monitor the reliability of the drive and predict drive failures, and to carry out different types of drive self tests. SMART Disk Monitoring Daemon (smartd) is enabled immediately after the insertion of a USB 3.0 SSD and starts logging warnings and errors in the /crashinfo/tracelogs/smart_errors.log. These warnings and errors are also displayed on the console. On removing the USB 3.0 SSD, smartd stops running.

USB 3.0 SSD is supported as a field-replaceable unit (FRU) that offers flexible storage configurations. If SSD is used initially on a PC, the default partition on USB 3.0 SSD is created by the PC supporting all the file systems. If SSD is used initially on the switch, one partition of the drive is created to support EXT4 file system.

File System on USB 3.0 SSD

USB 3.0 SSD is shipped as a raw device. When the device boots up, Cisco IOS software creates a partition with EXT4 as the default file system. However, the device supports all EXT-based file systems such as EXT2, EXT3, and EXT4. Non-EXT based file systems such as VFAT, NTFS, LVM, and so on are not supported.

The following file system operations are supported on the drive:

- Read
- Write
- Delete
- Copy
- Format

Password Authentication on USB 3.0 SSD

To protect the drive from unauthorized access, you must enable security on USB 3.0 SSD by setting a user password. A USB 3.0 SSD supports the following security states:

- Security disabled — User password has not been configured on the drive. This is the out-of-box state which is the default for any new drive.
- Security enabled — User password has been configured on the drive.
- Locked — Security is enabled and the drive is inaccessible.
- Unlocked — Security is enabled or disabled, but the drive is accessible.

You can configure password authentication using the CLI as well as programmable NETCONF/YANG method.

How to Configure USB 3.0 SSD

The following sections provide information about configuring USB 3.0 SSD:

Formatting USB 3.0 SSD

Use the **format usbflash1:** {**ext2** | **ext3** | **ext4** | **secure**} command to format the EXT file systems or the entire drive.

To format the USB 3.0 SSD drive in a device stack, use **format usbflash1**-*switch_num***:** {**ext2** | **ext3** | **ext4** | **secure**} command.

Unmounting USB 3.0 SSD from a Switch or a Switch Stack

To safely remove the USB 3.0 SSD from a switch or a switch stack, use the **hw-module switch** <*switch_num*> **usbflash1 unmount** command in privileged EXEC mode. This command unmounts the filesystem created

upon insertion, and notifies the system to complete pending read or write operations, if any, to safely remove the drive from the switch.

```
Device#hw-module switch 1 usbflash1 unmount

      *Jan  5 22:21:32.723: %IOSXE-0-PLATFORM: Switch 1 R0/0: SSD_UNMOUNT_LOG: usbflash1:
has been unmounted. All the usbflash1 entries in    IOS will now be cleared until the SSD
is plugged back into the switch.
      *Jan  5 22:21:32.729: %IOSD_INFRA-6-IFS_DEVICE_OIR: Device usbflash1 removed
```

After you run this command, you will not be able to access the USB anymore. To use the USB again reinsert it into the switch.

If you run the **hw-module switch** <*switch_num*> **usbflash1 unmount** command on a switch or switch stack without inserting the USB, the following error message is displayed.

```
Device#hw-module switch 1 usbflash1 unmount

*Jun 20 22:50:40.321:
ERROR: USB Not Present in this Slot 1
```

Enabling Password Security on USB 3.0 SSD

The password authentication feature enables you to configure security on a USB 3.0 SSD in order to protect the drive from unauthorized access and associated risks.To enable security on a USB 3.0 SSD, follow these steps to set a password on the drive.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device>enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**hw-module switch** *switch-number* **usbflash1 security enable password** *usb-password* **Example:** `Device#hw-module switch 1 usbflash1 security enable password 1234`	Configures a user-defined password on the USB 3.0 SSD. **Note** Password security will take effect only after after Online Insertion and Removal (OIR) of the USB or a switch reload.

After Online Insertion and Removal (OIR) of the USB or a switch reload, the USB will be in *Enabled and Locked* state. To unlock and access the USB, you must configure the switch to use the USB 3.0 SSD password that you create in this task.

What to do next

To configure the USB 3.0 SSD password on the switch, see Configuring USB 3.0 SSD Password on a Switch, on page 152.

Configuring USB 3.0 SSD Password on a Switch

To access a password protected USB 3.0 SSD using a switch, you must configure the same USB 3.0 SSD password on the switch. USB 3.0 SSD will be in locked state after a switch reset or OIR of the drive. To unlock and access the drive, the switch prompts you to enter the USB 3.0 SSD password saved on the switch. This procedure saves the password to the running configuration on the switch in type-6 encryption format.

Using the Encrypted Preshared Key feature, you can securely store plain text passwords in type-6 format in NVRAM using the command-line interface (CLI). Type-6 passwords are encrypted. Although the encrypted passwords can be seen or retrieved, it is difficult to decrypt them to find out the actual password.

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	(Optional) **key config-key password-encrypt** *password* **Example:** `Device(config)#key config-key password-encrypt 123456789`	Configures the master key on the switch. The password configured using this command is the master encryption key that is used to encrypt all the other keys in the switch. **Note** Skip this step if you have already configured the master key on the switch.
Step 4	[**no**] **hw-module switch** *switch-number* **usbflash1-password** *usb-password* **Example:** `Device(config)#hw-module switch 1 usbflash1-password 1234`	**Note** Ensure the password matches the one that you have configured on the USB 3.0 SSD to enable security. Encrypts the password internally using type-6 encryption. Use the **no** form of the command to remove the USB 3.0 SSD password from the running configuration of the switch.
Step 5	**end** **Example:** `Device(config)#end`	Returns to privileged EXEC mode.

Unlocking USB 3.0 SSD

Follow these steps to unlock a USB 3.0 SSD:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**hw-module switch** *switch-number* **usbflash1 security unlock password** *usb-password* **Example:** `Device#hw-module switch 1 usbflash1 security unlock password 1234`	Unlocks the drive and makes the drive available for temporary access. Note that password security is still enabled on the drive and if you insert the drive on any other switch, the drive will be in locked state.

Disabling Password Security on USB 3.0 SSD

Follow these steps to disable security or to change the password configured on a USB 3.0 SSD.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**hw-module switch** *switch-number* **usbflash1 security disable password** *usb-password* **Example:** `Device #hw-module switch 1 usbflash1 security disable password 1234`	Disables security on USB 3.0 SSD and makes the drive accessible. You do not have to reload the switch or perform OIR of the drive for the changes to take effect. **Note** On a switch stack, enter the switch number of the switch on which you have inserted the USB 3.0 SSD.

Monitoring USB 3.0 SSD

You can view the contents of the USB 3.0 SSD before working on its contents. For example, before copying a new configuration file, you might want to verify that the filesystem does not already contain a configuration

file with the same name. To display information about files on a filesystem, use one of the privileged EXEC commands listed in the following table:

Table 22: Commands to Display Files on a Filesystem

Command Name	Description
dir usbflash1:	Displays the list of files on the USB flash filesystem on an active switch. To access flash partitions of a standby switch or the device members in a stack, use **usbflash1**–*n* where *n*, is the standby switch number or the stack member number.
dir usbflash1-*switch_num*:	Displays the list of files on the filesystem in a stack setup.
dir stby-usbflash1:	Displays the list of files on the filesystem on the standby switch in a stack setup.
show usbflash1: filesystem	Displays more information about the filesystem.
show inventory	Displays the physical inventory information for the USB hardware. After multiple switchovers, the **show inventory** command output might display the USB flash filesystem (usbflash1) for the active switch with the switch number. **Note** The **show inventory** command displays ”usbflash1” in the output only when the device is in “Disabled and Unlocked” state or “Enabled and Unlocked” state.
more `file-url`	Displays the logs with SMART errors and overall health of the drive.
show hw-module usbflash1 security status	Displays USB 3.0 SSD authentication status.

Troubleshooting Tips

The following sections provide troubleshooting tips:

Troubleshooting USB 3.0 SSD Insertion and Removal

Table 23: Errors and Troubleshooting

Error That You May Encounter	Troubleshooting
USB3.0 SSD not detected after insertion	• Check if you are using a Cisco USB 3.0 SSD. If not, remove the drive from the device, and replace it with a Cisco USB 3.0 SSD. • If you are using a Cisco USB 3.0 SSD and the system is unable to detect the drive, remove and reinsert the USB 3.0 SSD. If it continues to fail, the USB might be defective.

Error messages displayed on the console after removing USB 3.0 SSD:
`*Mar 20 00:48:16.353: %IOSXE-4-PLATFORM: Switch 1 R0/0: kernel: xhci_hcd 0000:00:14.0: Cannot set link state.`
`*Mar 20 00:48:16.353: %IOSXE-3-PLATFORM: Switch 1 R0/0: kernel: usb usb4-port1: cannot disable (err = -32)`
`*May 10 01:12:49.603: %IOSXE-3-PLATFORM: Switch 3 R0/0: kernel: JBD2: Error -5 detected when updating journal superblock for sda1-8.`
	Remove the USB 3.0 SSD from the device after running the **unmount** command. For more information, see Unmounting USB 3.0 SSD from a Switch or a Switch Stack, on page 150.

Error message displayed on the console on inserting a non-Cisco USB 3.0 SSD: `%IOSXEBOOT-4-SSD_MOUNT_LOG: (local/local): ***INFO: Not a CISCO SSD - Cannot be used***`	Remove the USB from the device, and replace it with a Cisco USB 3.0 SSD.

Troubleshooting Password Authentication

Table 24: Errors and Troubleshooting

Error That You May Encounter	Troubleshooting
USB3.0 SSD not detected after insertion	Run the **show hw-module usbflash1 security status** command and check for `USB Authentication Status` fields in the output. If the `USB Authentication Status` field in the output displays `Enabled and Locked`, perform one of the following: • Unlock the drive temporarily using the **hw-module switch 1** *switch-number* **usbflash1 security unlock password** *usb-password* command. • Configure USB 3.0 SSD password on the switch. See Configuring USB 3.0 SSD Password on a Switch, on page 152.

USB 3.0 SSD password does not match the password saved in the running configuration of the switch. The switch displays the following error messages:
`*Oct 19 19:32:04.094: %IOSD_INFRA-6-IFS_DEVICE_OIR: Device usbflash1 added`
`*Oct 19 19:32:04.138: Warning: Configured password on SWITCH does not match with that on DRIVE.`
`Please remove password from SWITCH first and then from DRIVE to re-configure.`
	Perform the following:
	• Remove the password from the switch and reconfigure the switch to use the correct password. See Configuring USB 3.0 SSD Password on a Switch, on page 152.

USB 3.0 SSD without a password inserted on a switch that has the drive password configured. An attempt to unlock the disk using the password configured on the switch fails and the switch displays the following messages:
`*Dec 14 00:01:00.374: %IOSD_INFRA-6-IFS_DEVICE_OIR: Device usbflash1 added`
`*Dec 14 00:01:00.430: ERROR: No password configured on DRIVE. Remove password from SWITCH to re-configure.`
	Perform the following steps:
	1. Enable security on the drive USB 3.0 SSD. See Enabling Password Security on USB 3.0 SSD, on page 151.
	2. Reconfigure the password on the switch. See Configuring USB 3.0 SSD Password on a Switch, on page 152.

Error That You May Encounter	Troubleshooting

USB 3.0 SSD configured with a password inserted on a switch that does not have the drive password configured. An attempt to unlock the disk fails and the switch displays the following messages:
`*Oct 19 19:36:18.003: %IOSD_INFRA-6-IFS_DEVICE_OIR: Device usbflash1 added`
`*Oct 19 19:36:18.028: Warning: No password configured on SWITCH. Remove password from DRIVE to re-configure`
	Do one of the following:
	• Disable the password configured on the drive. See Disabling Password Security on USB 3.0 SSD, on page 153.
	• Configure password on the switch. See Configuring USB 3.0 SSD Password on a Switch, on page 152.

A USB 3.0 SSD in `Disabled and locked` state indicates that the USB drive has become unusable because of corrupted hardware.	To unlock and enable the drive, contact TAC.

Configuration Examples for USB 3.0 SSD

The following sections provide configuration examples for USB 3.0 SSD:

Example: Displaying USB 3.0 SSD Authentication Status

This example shows the USB 3.0 SSD authentication status on a switch stack with 4 switches.

```
# show hw-module usbflash1 security status

Switch#   USB Authentication        Status
-----------------------------------------------------
1         USB Not Present          USB 3.0 is not present
2         Disabled and Unlocked    Security is disabled & the drive in unlocked state
(Default state if USB is  present)
3         Enabled and Locked       Security Enabled and the drive in locked state
4         Enabled and Unlocked     Security Enabled and the drive in unlocked state
```

When the drive is in *Enabled and Unlocked* or *Disabled and Unlocked* state, you can format a drive and perform normal file system operations like read, write, delete, and copy.

Examples: Verifying the Filesystem

The following example displays the output of the **dir usbflash1:/** command in privileged EXEC mode:

```
Switch#dir usbflash1:

Directory of usbflash1:/
11  drwx            16384   Oct 9 2015 01:49:18 +00:00  lost+found
3145729  drwx             4096   Oct 9 2015 04:10:41 +00:00  test
118014062592 bytes total (111933120512 bytes free)
```

The following example displays the output of the **dir usbflash1:***switch_num***:** command in a device stack:

```
Switch#dir  usbflash1-2:
Directory of usbflash1-2:/

11 drwx 16384 Jun 8 2018 21:35:39 +00:00 lost+found

118014083072 bytes total (111933390848 bytes free)
```

Alternately, you can use the **dir stby-usbflash1:** command to access the file system on a standby switch:

```
Switch#dir  stby-usbflash1:
Directory of usbflash1-3:/
11  drwx            16384  May 16 2018 23:32:43 +00:00  lost+found
118014083072 bytes total (110358429696 bytes free)
```

To display the file system information for usbflash1, use the **show usbflash1: filesystem** command in privileged EXEC mode:

```
Switch#show usbflash1: filesystem
Filesystem: usbflash1
Filesystem Path: /vol/usb1
Filesystem Type: ext4
```

Examples: Verifying Physical Inventory Information

To display the physical inventory information for USB 3.0 SSD hardware, use the **show inventory** command:

```
Switch#show inventory

NAME: "usbflash1", DESCR: "usbflash1"
PID: SSD-240G          , VID: STP21460FN9, SN: V01
```

The following is a sample output of the **show inventory** command in a device stack:

```
Switch#show inventory

NAME: "usbflash1", DESCR: "usbflash1"
PID: SSD-240G          , VID: STP21460FN9, SN: V01

NAME: "usbflash1-3", DESCR: "usbflash1-3"
PID: SSD-240G          , VID: STP21310001, SN: V01
```

Examples: Verifying the Health of the Drive

To check the overall health of the drive, use the **more flash:smart_overall_health.log** command in privileged EXEC mode:

```
Switch#more flash:smart_overall_health.log

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

To check the health error logs, use the **more crashinfo:tracelogs/smart_errors.log** command in privileged EXEC mode:

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

Note The system might display warnings in the smart_errors.log. You can ignore these if the overall health self assessment in the flash/smart_overall_health.log displays `PASSED`.

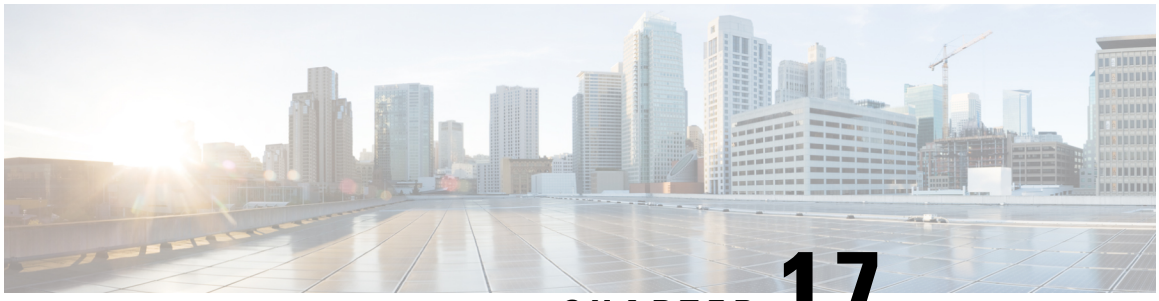
Feature History for USB 3.0 SSD

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

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

Release	Feature	Feature Information
Cisco IOS XE Fuji 16.9.1	USB 3.0 SSD	USB 3.0 SSD provides extra 120 GB storage to be used as a general-purpose storage device and as an application-hosting device.
Cisco IOS XE Fuji 16.9.6	USB 3.0 SSD storage	USB 3.0 SSD storage capacity increased to 240 GB in Cisco IOS XE Fuji 16.9.6 and later Cisco IOS XE Fuji 16.9 releases.
Cisco IOS XE Gibraltar 16.10.1	Password authentication	Password authentication feature enables you set a password on the USB 3.0 SSD device in order to protect the drive from unauthorized access and associated risks.
Cisco IOS XE Gibraltar 16.12.4	USB 3.0 SSD storage	USB 3.0 SSD storage capacity increased to 240 GB in Cisco IOS XE Gibraltar 16.12.4 and later Cisco IOS XE Gibraltar 16.12 releases.
Cisco IOS XE Amsterdam 17.3.1	USB 3.0 SSD storage	USB 3.0 SSD storage capacity increased to 240 GB in Cisco IOS XE Amsterdam 17.3.1 and all later releases.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 17

Configuring an External USB Bluetooth Dongle

- Restrictions for Configuring an External USB Bluetooth Dongle , on page 161
- Information About External USB Bluetooth Dongle, on page 161
- How to Configure an External USB Bluetooth Dongle on a Switch, on page 162
- Verifying Bluetooth Settings on a Switch, on page 163
- Feature History for Configuring an External Bluetooth Dongle, on page 163

Restrictions for Configuring an External USB Bluetooth Dongle

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

Information About External USB Bluetooth Dongle

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

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

Supported External USB Bluetooth Dongle

The following external USB Bluetooth dongles are supported:

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

How to Configure an External USB Bluetooth Dongle on a Switch

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

Procedure

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

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

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

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

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

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

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

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

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

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

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

or

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

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

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

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

Verifying Bluetooth Settings on a Switch

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

Table 25: Commands to Monitor Bluetooth Settings on a Device

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

Feature History for Configuring an External Bluetooth Dongle

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

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

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.12.1	Configuring an External Bluetooth Dongle	External USB Bluetooth dongle acts as a Bluetooth host for external devices and serves as a management port on the switch.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.