Configuring Interface Characteristics

- Information About Interface Characteristics, on page 1
- How to Configure Interface Characteristics, on page 13
- Configuration Examples for Interface Characteristics, on page 30
- Additional References for Configuring Interface Characteristics, on page 33
- Feature History for Configuring Interface Characteristics, on page 33

Information About Interface Characteristics

The following sections provide information about interface characteristics.

Interface Types

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

Port-Based VLANs

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

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

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

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

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

Switch Ports

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

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

Access Ports

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

The types of access ports supported are:

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

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

Trunk Ports

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

Although by default, a trunk port is a member of every VLAN known to the VTP, you can limit VLAN membership by configuring an allowed list of VLANs for each trunk port. The list of allowed VLANs does not affect any other port but the associated trunk port. By default, all possible VLANs (VLAN ID 1 to 4094) are in the allowed list. A trunk port can become a member of a VLAN only if VTP knows of the VLAN and if the VLAN is in the enabled state. If VTP learns of a new, enabled VLAN and the VLAN is in the allowed list for a trunk port, the trunk port automatically becomes a member of that VLAN and traffic is forwarded to and from the trunk port for that VLAN. If VTP learns of a new, enabled VLAN that is not in the allowed list for a trunk port, the port does not become a member of the VLAN, and no traffic for the VLAN is forwarded to or from the port.

Tunnel Ports

Tunnel ports are used in IEEE 802.1Q tunneling to segregate the traffic of customers in a service-provider network from other customers who are using the same VLAN number. You configure an asymmetric link from a tunnel port on a service-provider edge switch to an IEEE 802.1Q trunk port on the customer switch.

Packets entering the tunnel port on the edge switch, already IEEE 802.1Q-tagged with the customer VLANs, are encapsulated with another layer of an IEEE 802.1Q tag (called the metro tag), containing a VLAN ID unique in the service-provider network, for each customer. The double-tagged packets go through the service-provider network keeping the original customer VLANs separate from those of other customers. At the outbound interface, also a tunnel port, the metro tag is removed, and the original VLAN numbers from the customer network are retrieved.

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

Routed Ports

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

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

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

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

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

Switch Virtual Interfaces

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

Note You cannot delete interface VLAN 1.

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

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

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

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

EtherChannel Port Groups

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

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

Network Modules

The following table shows the list of supported uplink ports:

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

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

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

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

Power over Ethernet

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

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

Using the Switch USB Ports

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

USB Mini-Type B Console Port

The device has the following console ports:

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

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

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

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

Console Port Change Logs

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

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

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

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

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

USB Type A Port

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

Disabling USB Ports

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

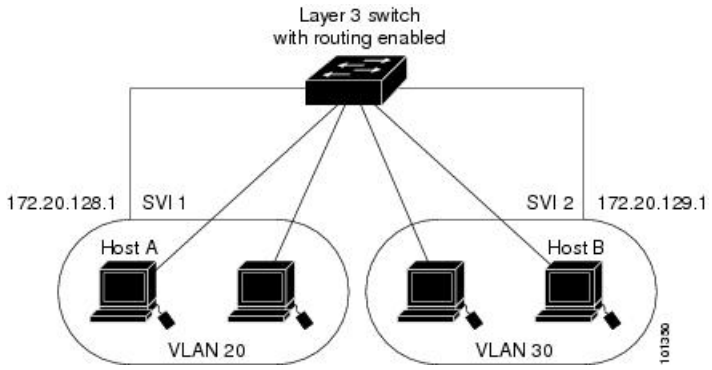
When a USB port is disabled, no system messages are generated if a USB is inserted.

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

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

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

These are examples of how to configure interfaces on standalone device:

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

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

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

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

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

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

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

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

Default Ethernet Interface Configuration

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

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

Table 1: Default Layer 2 Ethernet Interface Configuration

Feature	Default Setting
Operating mode	Layer 2 or switching mode (**switchport** command).
Allowed VLAN range	VLANs 1 to 4094.
Default VLAN (for access ports)	VLAN 1 (Layer 2 interfaces only).
Native VLAN (for IEEE 802.1Q trunks)	VLAN 1 (Layer 2 interfaces only).
VLAN trunking	Switchport mode dynamic auto (supports DTP) (Layer 2 interfaces only).
Port enable state	All ports are enabled.
Port description	None defined.
Speed	Autonegotiate.
Duplex mode	Autonegotiate.
Flow control	Flow control is set to **receive: on**. It is always off for sent packets.
EtherChannel (PAgP)	Disabled on all Ethernet ports.
Port blocking (unknown multicast and unknown unicast traffic)	Disabled (not blocked) (Layer 2 interfaces only).
Broadcast, multicast, and unicast storm control	Disabled.
Protected port	Disabled (Layer 2 interfaces only).
Port security	Disabled (Layer 2 interfaces only).
Port Fast	Disabled.

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

Interface Speed and Duplex Mode

Gigabit Ethernet interfaces on the switch operate at 10, 100, 1000 Mbps speed and in either full-duplex or half-duplex mode. In full-duplex mode, two stations can send and receive traffic at the same time. Normally, 10-Mbps ports operate in half-duplex mode, which means that stations can either receive or send traffic. The switch also includes multi-Gigabit Ethernet ports, which support speeds at 100 Mb, 1 Gb, 2.5 Gb, 5 Gb, and 10 Gb and operate at full-duplex mode, SFP modules that support speeds up to 1 Gbps, SFP+ modules that support speeds up to 10 Gbps, and SFP28 modules that support speeds up to 25 Gbps, and QSFP modules that support speeds up to 40 Gb/s. For the list of supported switch models, refer *Cisco Catalyst 9200 Series Switches Hardware Installation Guide*.

Speed and Duplex Configuration Guidelines

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

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

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

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

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

- If both ends of the line support autonegotiation, we highly recommend the default setting of **auto** negotiation.
- If one interface supports autonegotiation and the other end does not, configure duplex and speed on both interfaces; do not use the **auto** setting on the supported side.

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

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

Port Settings

The **port-settings** command can simultaneously or separately configure the speed, duplex, and auto negotiation for an interface, an interface range, or a port channel interface.

When using a single command to configure multiple parameters of the port-settings command, the order must be, **speed**, **duplex**, and **autoneg**. If you specify **speed** first, you can configure **duplex** and **autoneg** for the interface. If you specify **duplex** first, you can only configure **autoneg**. And, if you specify **autoneg** first, you cannot configure **speed** or **duplex**.

If the **port-settings** command is configured, this configuration is displayed only in the output of the **show running-config yang** command. The **show running-config** command output will not display information about this command.

For more information, see the section, Configuring Port Settings for an Interface.

Comparison of Commands

The **port-settings** command can be used instead of the **speed**, **duplex**, and **negotiation auto** commands available in the interface configuration mode.

The **port-settings** , **speed**, **duplex**, and **negotiation auto** commands coexist in the CLI.

Because the same parameters for an interface can be configured through two commands, the last configured values are used by the configuration.

This table provides a one-to-one comparison of the existing and newly-added commands:

Table 2: Command Comparison

Existing Command	Newly-Added port-settings Command
speed 10	**port-settings speed 10**
speed 100	**port-settings speed 100**
speed auto	**port-settings speed auto**
speed auto 10 100	**port-settings speed auto-list 10 100**
speed nonegotiate	**port-settings autoneg disable**

Existing Command	Newly-Added port-settings Command
duplex half	**port-settings duplex half**
duplex full	**port-settings duplex full**
negotiation auto	**port-settings autoneg enable**
no negotiation auto	**port-settings autoneg disable**

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

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

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

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

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

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

How to Configure Interface Characteristics

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

Configuring an Interface

These general instructions apply to all interface configuration processes.

Procedure

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

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

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

Adding a Description for an Interface

Follow these steps to add a description for an interface.

Procedure

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

Configuring a Range of Interfaces

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **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. **Note** Use the normal configuration commands to apply the configuration parameters to all interfaces in the range. Each command is executed as it is entered.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**show interfaces** [*interface-id*] **Example:** `Device# show interfaces`	Verifies the configuration of the interfaces in the range.

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

Configuring and Using Interface Range Macros

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**define interface-range** *macro_name interface-range* **Example:**	Defines the interface-range macro, and saves it in NVRAM. • The *macro_name* is a 32-character maximum character string. • A macro can contain up to five comma-separated interface ranges. • Each *interface-range* must consist of the same port type. **Note** Before you can use the **macro** keyword in the **interface range macro** global configuration command string, you must use the **define interface-range** global configuration command to define the macro.

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

Setting the Interface Speed and Duplex Parameters

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

Procedure

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

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

Configuring Port Settings for an Interface

Use the **port-settings** command to simultaneously or separately configure the speed, duplex, and auto negotiation for an interface.

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	**interface** *interface-name interface-ID*	Configures an interface, and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**port-settings speed** {**10** | **100** | **1000** | **auto** | **auto-list**}	Enters the appropriate speed parameter for the interface.
Step 5	**port-settings duplex** {**auto** | **full** | **half**}	Configures duplex operation on an interface.
Step 6	**port-settings autoneg** {**enable** | **disable**}	Configures duplex operation on an interface.
Step 7	**port-settings speed 1000 duplex full autoneg enable**	Configures port settings for an interface. • When configuring all port-setting parameters in a single command, the order must be **speed**, **duplex**, and **autoneg**.
Step 8	**end**	Exits interface configuration mode and returns to privileged EXEC mode.
Step 9	**show running-config yang**	Displays the YANG-specific configurations on the device. • The configured **port-settings** for the interface is displayed only in the output of the **show running-config yang** command.

Example

This example shows how to set the port settings for an interface.

```
Device# enable
Device# configure terminal
Device(config)# interface gigabitethernet 0/0
Device(config-if)# port-settings speed 1000
Device(config-if)# port-setting duplex full
Device(config-if)# port-setting autoneg enable
Device(config-if)# end
```

This example shows how to set all port setting parameters for an interface in a single command.

```
Device# enable
Device# configure terminal
Device(config)# interface gigabitethernet 0/0
Device(config-if)# port-settings speed 1000 duplex full autoneg enable
Device(config-if)# end
```

This sample output from the **show running-config yang** command, displays port settings configured for an interface.

```
Device# show running-config yang

Device# show running-config yang
Building configuration...
Current configuration : 19884 bytes
!!
Last configuration change at 23:53:04 UTC Tue Jun 11 2024 by platform
!
```

```
version 17.15
service timestamps debug datetime msec
service timestamps log datetime msec
!
hostname Device
!! vrf definition Mgmt-vrf
!
address-family ipv4
exit-address-family
!
address-family ipv6
exit-address-family
!
aaa new-model
!
aaa authentication login default local
aaa authorization exec default local
!
!
!
login on-success log
vtp domain GOV
vtp mode transparent
!!!!!!!
stackwise-virtual
 domain 1
!
crypto pki trustpoint SLA-TrustPoint
 enrollment pkcs12
 revocation-check crl
 hash sha256
!
!
!
interface Port-channel1
no shutdown
port-settings speed 1000 duplex full
!
interface GigabitEthernet0/0
vrf forwarding Mgmt-vrf
ip address 192.0.2.1 255.255.255.0
no shutdown
port-settings autoneg enable
!
interface GigabitEthernet1/1/0/1
no shutdown
!
line vty 0 4
exec-timeout 0 0
transport input ssh
line vty 5 15
transport input ssh
!
netconf-yang
end
Device#
```

Configuring the IEEE 802.3x Flow Control

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet1/0/1`	Specifies the physical interface to be configured, and enters interface configuration mode.
Step 4	**flowcontrol** {**receive**} {**on** | **off** | **desired**} **Example:** `Device(config-if)# flowcontrol receive on`	Configures the flow control mode for the port.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show flowcontrol interface** *interface-id* **Example:** `Device# show flowcontrol interface GigabitEthernet1/0/1`	Verifies the specified interface flow control settings.
Step 7	**show flowcontrol module** *slot* **Example:** `Device# show flowcontrol module 1`	Verifies the interface flow control settings on the module.
Step 8	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring a Layer 3 Interface

Follow these steps to configure a layer 3 interface.

Procedure

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

	Command or Action	Purpose
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.

Because outer GRE packet cannot be fragmented, when GRE is configured on a switch, the maximum transmission unit (MTU) configured on the tunnel must be honored on the underlay physical network. This means that all links in the path from one end of the tunnel to the other must be equal to or grater than the tunnel MTU.

Note

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

To configure a GRE tunnel, perform this task:

Procedure

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

	Command or Action	Purpose
Step 3	**interface tunnel** *number* **Example:** `Device(config)# interface tunnel 2`	Enables tunneling on the interface.
Step 4	**ip address** *ip_address subnet_mask* **Example:** `Device(config)# ip address 100.1.1.1 255.255.255.0`	Configures the IP address and IP subnet.
Step 5	**tunnel source** {*ip_address* | *type_number*} **Example:** `Device(config)# tunnel source 10.10.10.1`	Configures the tunnel source.
Step 6	**tunnel destination** {*host_name* | *ip_address*} **Example:** `Device(config)# tunnel destination 10.10.10.2`	Configures the tunnel destination.
Step 7	**tunnel mode gre ip** **Example:** `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.

	Command or Action	Purpose
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet1/0/2`	Specifies a Layer 2 interface (physical port or port channel), and enters interface configuration mode.
Step 4	**switchport autostate exclude** **Example:** `Device(config-if)# switchport autostate exclude`	Excludes the access or trunk port when defining the status of an SVI line state (up or down)
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show running config interface** *interface-id* **Example:** `Device# show running config interface gigabitethernet1/0/2`	(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:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**interface** {**vlan** *vlan-id*} | { **gigabitethernet** *interface-id*} | {**port-channel** *port-channel-number*} **Example:** `Device(config)# interface gigabitethernet1/0/2`	Selects the interface to be configured.
Step 4	**shutdown** **Example:** `Device(config-if)# shutdown`	Shuts down an interface.
Step 5	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Restarts an interface.
Step 6	**end** **Example:** `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.

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	**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

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

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**line console 0** **Example:**	Configures the console and enters line configuration mode.

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

Disabling USB Ports

To disable all USB ports, peform this procedure.

Procedure

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

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

Monitoring Interface Characteristics

The following sections provide information about monitoring interface characteristics.

Monitoring Interface Status

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

Table 3: show Commands for Interfaces

Command	Purpose
show interfaces *interface-id* **status** [**err-disabled**]	Displays interface status or a list of interfaces in the error-disabled state.
show interfaces [*interface-id*] **switchport**	Displays administrative and operational status of switching (nonrouting) ports. You can use this command to find out if a port is in routing or in switching mode.
show interfaces [*interface-id*] **description**	Displays the description configured on an interface or all interfaces and the interface status.
show ip interface [*interface-id*]	Displays the usability status of all interfaces configured for IP routing or the specified interface.
show interface [*interface-id*] **stats**	Displays the input and output packets by the switching path for the interface.
show interface [*interface-id*] **link**[**module** *number*]	Displays the up time and down time of an interface or all interfaces.
show interfaces *interface-id*	(Optional) Displays speed and duplex on the interface.
show interfaces transceiver dom-supported-list	(Optional) Displays Digital Optical Monitoring (DOM) status on the connect SFP modules.
show interfaces transceiver properties	(Optional) Displays temperature, voltage, or amount of current on the interface.
show interfaces [*interface-id*] [{**transceiver properties** | **detail**}] *module number*]	Displays physical and operational status about an SFP module.
show running-config interface [*interface-id*]	Displays the running configuration in RAM for the interface.
show version	Displays the hardware configuration, software version, the names and sources of configuration files, and the boot images.

Command	Purpose
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 4: 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 enter interface-range configuration mode for the interface-range macro *enet_list*:

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

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

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

Example: Setting Interface Speed and Duplex Mode

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

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

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

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

Example: Configuring a Layer 3 Interface

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

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

Example: Configuring the Console Media Type

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

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

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

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

Example: Configuring USB Inactivity Timeout

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

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

The following example shows how to disable the configuration:

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

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

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

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

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

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

Additional References for Configuring Interface Characteristics

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	See the "Interface and Hardware Commands" section in the *Command Reference (Catalyst 9200 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 Fuji 16.9.2	Interface Characteristics	Interface Characteristics includes interface types, connections, configuration modes, speed, and other aspects of configuring a physical interface on a device.
Cisco IOS XE Gibraltar 16.11.1	Multi-Gigabit Ethernet Interfaces	Support for Multi-Gigabit Ethernet ports operating at 100Mb/s, 1Gb/s, 2.5 Gb/s, 5Gb/s, and 10 Gb/s was introduced on all the models of the series
Cisco IOS XE Bengaluru 17.5.1	Disabling USB interfaces	Support to disable all USB ports on a standalone or stacked device was introduced.

Release	Feature	Feature Information
Cisco IOS XE Cupertino 17.9.1	Interface Characteristics Disabling USB interfaces Multi-Gigabit Ethernet Interfaces	This feature was implemented on the C9200CX-12P-2X2G, C9200CX-8P-2X2G, and C9200CX-12T-2X2G models.

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