

Revised: September 25, 2025

Interface Characteristics Configuration Guide

Feature history for interface characteristics

This table provides release and platform support 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 name and description	Supported platform
Cisco IOS XE 17.18.1	Interface characteristics: Interface characteristics includes interface types, connections, configuration modes, speed, and other aspects of configuring a physical interface on a device.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

Understand interface characteristics

These sections provide information about interface characteristics.

Interface types

This section describes the different types of interfaces supported on the device.

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,
- forwards packets received on a port only to ports belonging to the same VLAN, providing a hard firewall for traffic, and
- requires a Layer 3 device to enable communication between different VLANs.

VLAN partitions provide hard traffic barriers within the VLAN, and each VLAN has its own MAC address table.

Port-based VLAN configuration

You can create a VLAN by associating the port with the VLAN through local port configuration, by using the VLAN Trunking Protocol (VTP) to learn from a neighbor on a trunk, or by manually creating it. VLANs can be formed with ports across the stack.

Configure VLANs by using 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. Set VTP mode to transparent before configuring extended-range VLANs (VLAN IDs 1006 to 4094), if VTP version is 1 or 2. Extended-range VLANs created in transparent mode are not added to the VLAN database but are saved in the running configuration. With VTP version 3, you can create extended-range VLANs in client or server mode in addition to transparent mode. These VLANs are saved in the VLAN database.

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

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

- For an access port, set and define the VLAN to which it belongs.

Layer 2 interfaces

Layer 2 interfaces on switch devices are switch ports associated with a physical port and assigned to one or more VLANs. A switch port can be an access port, a trunk port, or a tunnel port. You can configure a port as an access port or trunk port. 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

The access port carries traffic for only one VLAN, except when configured in a voice VLAN port. It receives and sends traffic 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 is not configured with a voice VLAN, packets received with a tag are dropped. If an access port is configured with a voice VLAN, packets tagged with the voice VLAN are accepted, and packets tagged with any other VLANs are dropped.

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

You can configure an access port with an attached Cisco IP Phone to use one VLAN for voice traffic and another for data traffic.

Trunk ports

A trunk port is a network port that:

- allows communication between multiple VLANs,
- supports both tagged and untagged traffic using IEEE 802.1Q standards, and
- enables VLAN membership through VLAN Trunking Protocol (VTP) and an allowed VLAN list.

A trunk port configured with IEEE 802.1Q tagging can receive both tagged and untagged traffic. By default, the device forwards untagged traffic in the native VLAN configured for the port. The native VLAN is VLAN 1 by default. The native VLAN can be assigned any VLAN ID. If a packet has a VLAN ID that is the same as the outgoing port native VLAN ID, the packet is sent untagged; otherwise, the device sends the packet with a tag.

Trunk ports are members of every VLAN known to VTP by default. However, you can limit their VLAN membership using an allowed list 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 the VLAN is enabled. 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 components used in IEEE 802.1Q tunneling that:

- segregate customer traffic in a service-provider network from other customers using the same VLAN number,
- encapsulate packets with a unique VLAN ID for each customer in a service-provider network, and
- allow the retrieval of original customer VLAN numbers after crossing the service-provider network.

Tunnel ports are configured asymmetrically from a tunnel port on a provider's edge switch to a trunk port on a customer's switch. They must belong to a VLAN unique to each customer and cannot be trunk or access ports.

Layer 3 interfaces

A Layer 3 device can have an IP address assigned to each routed port and SVI. You can configure a maximum of 4000 Layer 3 interfaces.

If the device is using its maximum hardware resources, attempts to create a routed port or SVI have these results:

- If you try to create a new routed port, the device generates a message indicating insufficient resources to convert the interface. Consequently, the interface remains 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 indicates the VLAN in a suspended state.
- If the device configuration includes more virtual interfaces, such as SVIs or VLAN subinterfaces, than the hardware can support, Layer 2 VLANs are created, but the SVIs or subinterfaces that exceed support remain down. The device sends a message indicating insufficient hardware resources.



Note

- All Layer 3 interfaces require an IP address to route traffic.
- 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.

Routed ports

A routed port is a physical port that acts like a router port and can operate independently from a router. A routed port is not associated with a particular VLAN, as is an access port.

Routed ports can be configured with a Layer 3 routing protocol. A routed port is a Layer 3 interface and does not support Layer 2 protocols, such as DTP and STP.

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

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, and are supported for routed port-channels. A subinterface can support IP addressing, forwarding, QoS, 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.



Note

Do not use VLAN 1 as encapsulation 802.1Q VLAN.

Switch virtual interfaces

A Switch Virtual Interface (SVI) is a virtual Layer 3 interface representing a VLAN. It provides Layer 3 processing for packets to and from all switch ports associated with that VLAN. You can associate only one SVI with a VLAN. Configure an SVI either to route between VLANs or to provide IP host connectivity to the device. By default, an SVI is created for the default VLAN 1 to allow remote device administration. Additional SVIs must be explicitly configured. SVIs are created by entering a VLAN ID after the **interface vlan** global configuration command.



You cannot delete interface VLAN 1.

SVI configuration

SVIs provide IP host connectivity only to the system. They 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. After creating the VLAN interface, you can use the **interface range vlan id** command to configure the VLAN interface.

The device supports a total of 4094 VLANs and SVIs. However, the interplay between the number of SVIs, routed ports, and other configured features might impact CPU performance because of hardware limitations.

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.



-
- An SVI becomes active only after it is associated with a physical port.
 - SVI MAC addresses remain unchanged after a device reload, which is expected.
-

EtherChannel port groups

EtherChannel port groups combine 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.

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, multiple access ports into one logical access port, multiple tunnel ports into one logical tunnel port, or multiple routed ports into one logical routed port.

Most protocols operate over either single physical ports or aggregated switch ports and do not recognize the physical ports within the port group. Exceptions like DTP, Cisco Discovery Protocol (CDP), and Port Aggregation Protocol (PAgP) operate only on physical ports.

EtherChannel configuration

When you configure 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.

Layer 2 port-channel supports SVIs, and Layer 3 port-channel supports subinterfaces.

Interfaces by speed

Different speeds are supported depending on the type of interfaces used on the device.

Multigigabit Ethernet

The MultiGigabit Ethernet (mGig) feature lets you configure speeds of 10 Mbps, 100 Mbps, 1 Gbps, 2.5 Gbps, 5 Gbps, and 10 Gbps with automatic bandwidth negotiation over traditional CAT5e cables and higher cable variants.

These Cisco C9350 Series Smart Switches support the mGig feature:

- C9350-48HX
- C9350-48TX
- C9350-48U
- C9350-24U

Cisco C9610 Series Smart Switches supports 10 Gigabit Ethernet interfaces on C9600-LC-48TX line card.

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. This occurs when a higher-speed link cannot be established or when the quality of an established link degrades to a level that requires the PHY to reestablish it. Recommended downshift speed values are:

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

Interface speed and duplex mode



Note

Duplex mode is not supported on Cisco C9610 Series Smart Switches.

Ethernet interfaces on the switch operate at 10 Mbps, 100 Mbps, 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.

The 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, 5 Gbps, and 10 Gbps; 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.

Speed and duplex configuration guidelines

Follow these guidelines when configuring interface speed and duplex mode:

- 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 (with port fast disabled) 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. Configure speed and duplex options on a link to auto or fixed on both ends for best results. 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 may cause the interface to shut down and restart during reconfiguration.

Port settings

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

When configuring multiple port-settings parameters using a single command, follow this order: **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**.
- If you specify **autoneg** first, you cannot configure **speed** or **duplex**.

When the **port-settings** command is configured, its configuration appears only in the output of the **show running-config yang** command, and not in the **show running-config** command output.

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.

The configuration uses the last configured values when the same interface parameters are set using two commands. The table below offers a detailed comparison of existing commands and newly added commands.

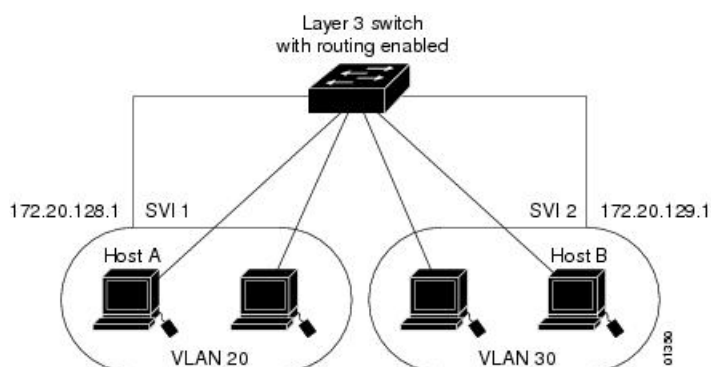
Table 1: Command comparison

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
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

Interface connections

Devices within a single VLAN can communicate directly through any switch. Ports in different VLANs must use a routing device to exchange data. A standard Layer 2 device requires ports in different VLANs to exchange information through a router. Enabling routing on the device and configuring both VLAN 20 and VLAN 30 with an SVI that has an assigned IP address allows packets to be sent directly from Host A to Host B through the device, eliminating the 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. The device hardware forwards traffic whenever possible to maintain high performance. However, only IPv4 packets with Ethernet II encapsulation are routed within the hardware.

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

Interface configuration mode

You can also configure a range of interfaces. An interface on the device is represented by a 3-tuple notation that lists the module, subslot, and port.

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

- Type:
 - **Cisco C9350 Series Smart Switches:** 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, 40-Gigabit Ethernet (FortyGigabitEthernet or fo) for 40 Gbps, 50-Gigabit Ethernet (FiftyGigabitEthernet or fif) for 50 Gbps, and 100-Gigabit Ethernet (HundredGigE or hu) for 100Gbps.
 - **Cisco C96100 Series Smart Switches:** 10-Gigabit Ethernet (TenGigabitEthernet or te) for 10 Gbps, 25-Gigabit Ethernet (TwentyFiveGigE or twe) for 25 Gbps, 40-Gigabit Ethernet (FortyGigabitEthernet or fo) for 40 Gbps, 50-Gigabit Ethernet (FiftyGigabitEthernet or fif) for 50 Gbps, 100-Gigabit Ethernet (HundredGigE or hu) for 100Gbps, and 400-Gigabit Ethernet (FourHundredGigE or fou) for 400 Gbps.
- Module number: The module or slot number on the device. You can use the switch port LEDs, in stack mode, to identify the stack member number of a device.
- Port number: The interface number on the device. The port numbering starts with the far left port when facing the front of the device, for example, FortyGigabitEthernet1/0/1.

Subslot number: The subslot number is always 0.

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.

Here is an example of how to configure 10-Gigabit Ethernet port 1 on a standalone device:

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

Default Ethernet interface configuration

To configure Layer 2 parameters, enter the **switchport** command without parameters to switch the interface from Layer 3 mode to Layer 2 mode. Switching modes shuts down and then re-enables the interface, possibly generating messages on connected devices. Note that switching modes might cause the loss of previous configuration settings, returning the interface to its default state.

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

Table 2: 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).

Feature	Default Setting
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.
Auto-MDIX	Enabled.
Power over Ethernet (PoE)	Enabled (auto).

Interface features

This section describes various features that are supported on an interface.

Network modules



Note

Network modules are supported only on the Cisco C9350 Series Smart Switches.

Network modules are hardware components that expand functionality and connectivity options by providing additional network interfaces, services, or capabilities.

The device supports network modules that include 1-Gigabit Ethernet, 10-Gigabit Ethernet, 25-Gigabit Ethernet, 40-Gigabit Ethernet, 50-Gigabit Ethernet, and 100-Gigabit Ethernet uplink ports.

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.

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 these console ports:

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

While both ports provide console output, only one port is active for console input at any given 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 for connecting a PC or another device, which must include a terminal emulation application. When the device detects a valid USB connection to a powered-on device that supports host functionality (such as a PC), input from the RJ-45 console is immediately disabled, and input from the USB console is enabled. Removing the USB connection immediately reenables input from the RJ-45 console connection.

Console port change logs

When the software starts, 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. The first log from the device shows the RJ-45 console because the bootloader did not change to the USB console. A short time later, the console changes, and the USB console log appears.

```
switch-1
```

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

The hardware automatically switches to the RJ-45 console interface upon removal of the USB cable or deactivation of the USB connection by the PC. Configure the console type to remain as RJ-45, or set 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, including port densities of 128 MB, 256 MB, 1 GB, 4 GB, 8 GB, and 16 GB. Use standard Cisco IOS command line interface (CLI) commands to read, write, erase, and copy to or from the flash device. Configure the devices to boot from the USB flash drive.

Disable USB ports

You can disable all USB ports in a standalone or stacked device by using the **platform usb disable** command. To reenable the USB ports, use the **no platform usb disable** command.

System messages are not generated when a USB is inserted into a disabled port.



Note

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

This command works on a device configured with Cisco StackWise Virtual and Quad-Supervisor with Route Processor Redundancy.

IEEE 802.3x flow control

Flow control allows congested Ethernet ports to pause link operations, enabling control of traffic rates during congestion. When a port is congested and cannot receive more traffic, it sends a pause frame to notify the other port to stop sending until the condition clears. Upon receipt of a pause frame, the sending device halts the transmission of any data packets, which prevents any loss of data packets during the congestion period.



Note

While switch ports can receive pause frames, they cannot send them.

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.

Configure interface characteristics

This section provides information about the various tasks to configure interface characteristics.

Configure an interface

These general instructions apply to all interface configuration processes.

	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 <i>interface-id</i> Example: Device(config) # interface tengigabitethernet1/1/1 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.
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. • You must use the define interface-range global configuration command to define the macro before configuring the macro keyword in the interface range global configuration command string.
Step 6	show interfaces Example: Device(config) # 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.

Add a description for an interface

Follow these steps to add a description for an interface.

	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 <i>interface-id</i> Example: Device(config)# interface tengigabitethernet1/1/1	Specifies the interface for which you are adding a description, and enters interface configuration mode.
Step 4	description <i>string</i> 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 <i>interface-id</i> description Example: Device(config)# show interfaces tengigabitethernet1/1/1 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.

Configure a range of interfaces

Use the **interface range** global configuration command to configure multiple interfaces with the same configuration parameters. 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.

	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 range { <i>port-range</i> macro <i>macro_name</i> } Example: <pre>Device(config) # interface range macro</pre>	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 <i>port-range</i>, you must enter the interface type for each entry and enter spaces before and after the comma. - In a hyphen-separated <i>port-range</i>, 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: <pre>Device(config) # end</pre>	Returns to privileged EXEC mode.
Step 5	show interfaces [<i>interface-id</i>] Example: <pre>Device# show interfaces</pre>	Verifies the configuration of the interfaces in the range.
Step 6	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Configure and use interface range macros

Create an interface range macro to automatically select a range of interfaces for configuration.

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

	Command or Action	Purpose
Step 2	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 3	define interface-range <i>macro_name</i> interface-range Example: <pre>Device(config)# define interface-range enet_list tengigabitethernet1/1/1 - 2</pre>	Defines the interface-range macro, and saves it in NVRAM. • The <i>macro_name</i> is a 32-character maximum character string. • A macro can contain up to five comma-separated interface ranges. • Each <i>interface-range</i> must consist of the same port type.  Note You must use the define interface-range global configuration command to define the macro before configuring the macro keyword in the interface range global configuration command string.
Step 4	interface range macro <i>macro_name</i> Example: <pre>Device(config)# interface range macro enet_list</pre>	Selects the interface range to be configured using the values saved in the interface-range macro called <i>macro_name</i> .
Step 5	end Example: <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.
Step 6	show running-config include define Example: <pre>Device# show running-config include define</pre>	Shows the defined interface range macro configuration.
Step 7	copy running-config startup-config Example: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Shut down and restart an interface

Shutting down an interface disables all functions, making it unavailable on monitoring displays, and this status is communicated to other network servers through dynamic routing protocols. The interface is not mentioned in any routing updates.

	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} { tengigabitethernet interface-id} {port-channel port-channel-number} Example: Device(config)# interface tengigabitethernet1/1/1	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.

Configure a Layer 3 interface

Follow these steps to configure a layer 3 interface.

	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 {tengigabitethernet interface-id} {vlan vlan-id} {port-channel port-channel-number} Example: Device(config)# interface tengigabitethernet1/1/1	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.

Configure a logical Layer 3 GRE tunnel interface

Generic Routing Encapsulation (GRE) is a tunneling protocol used to encapsulate network layer protocols inside virtual point-to-point links. The GRE tunnel encapsulates the data without encrypting it.

 Note

- 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. Use the **ip mtu 256** command to set the lowest MTU on both ends of the GRE tunnel, preventing fragmentation.

Configure a GRE tunnel with these steps:

	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 <i>number</i> Example: Device(config)# interface tunnel 2	Enables tunneling on the interface.
Step 4	ip address <i>ip_address subnet_mask</i> Example: Device(config)# ip address 100.1.1.1 255.255.255.0	Configures the IP address and IP subnet.
Step 5	tunnel source {<i>ip_address</i> <i>type_number</i>} Example: Device(config)# tunnel source 10.10.10.1	Configures the tunnel source.
Step 6	tunnel destination {<i>host_name</i> <i>ip_address</i>} 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.

Configure SVI autostate exclude

Follow these steps to exclude SVI autostate.

	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 tengigabitethernet1/1/1	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 tengigabitethernet1/1/1	(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.

Set the interface speed and duplex parameters

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

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

	Command or Action	Purpose
	Example: Device> enable	Enter your password, if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface interface-id Example: Device(config)# interface tengigabitethernet1/1/3	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. • Enter auto to enable the interface to autonegotiate speed with the connected device. If you specify a speed and also set the auto keyword, the port autonegotiates only at the specified speeds. • The nonegotiate keyword is available only for SFP module ports. SFP module ports operate only at 1000 Mb/s but can be configured to not negotiate if connected to a device that does not support autonegotiation.
Step 5	duplex {auto full half} Example: Device(config-if)# duplex half	Enters the duplex parameter for the interface. 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 tengigabitethernet1/1/3	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.

Configure port settings for an interface

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

	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 <i>interface-name interface-ID</i>	Configures an interface, and enters interface configuration mode.
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 autonegotiate 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.

Configure 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.

	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.

Configure 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.

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

	Command or Action	Purpose
Step 4	usb-inactivity-timeout switch <i>switch_number</i> <i>timeout-minutes</i> Example: <pre>Device(config-line)# usb-inactivity-timeout switch 1 30</pre>	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: <pre>Device# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

Disable USB ports

To disable all USB ports, perform this procedure.

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

Configure the IEEE 802.3x flow control

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

	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 <i>interface-id</i> Example: Device(config)# interface tengigabitethernet1/1/1	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 <i>interface-id</i> Example: Device# show flowcontrol interface tengigabitethernet1/1/1	Verifies the specified 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.

Configuration examples

These sections provide examples of interface characteristics configurations.

Example: Add a description to an interface

This 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 tengigabitethernet1/1/1
Device(config-if)# description Connects to Marketing
Device(config-if)# end
Device# show interfaces tengigabitethernet1/1/1 description

```

```

Interface Status          Protocol Description
tel1/1/1    admin down    down      Connects to Marketing

```

Example: Configuring a range of interfaces

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

```

Device# configure terminal
Device(config)# interface range tengigabitethernet 1/1/1-2
Device(config-if-range)# shut

```

This example shows how to use a comma to add different interface type strings to the range to enable 10-Gigabit Ethernet ports 1 to 3 and 25-Gigabit Ethernet ports 1 and 2 to receive flow-control pause frames:

```

Device# configure terminal
Device(config)# interface range tengigabitethernet1/1/1 - 3 , twentyfivegig1/1/1 - 2
Device(config-if-range)# flowcontrol receive on

```



Note

If you enter multiple configuration commands in interface-range mode, the system executes each command as you enter it. Commands are not executed all at once after exiting interface-range mode. Exiting the mode while commands are being executed may result in some commands not being applied to all interfaces in the range. Exit interface-range configuration mode when the command prompt reappears.

Example: Configuring and using interface range macros

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

```

Device# configure terminal
Device(config)# define interface-range enet_list tengigabitethernet1/1/1 - 2
Device(config)# end
Device# show running-config | include define
define interface-range enet_list tengigabitethernet1/1/1 - 2

```

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

```

Device# configure terminal
Device(config)# define interface-range macro1 tengigabitethernet1/1/1 - 2, tengigabitethernet1/1/1 - 7,
twentyfivegig1/1/1 - 2
Device(config)# end

```

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

```

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

```

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

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

Example: Configure a Layer 3 interface

This 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 tengigabitethernet1/1/1
Device(config-if)# no switchport
Device(config-if)# ip address 192.20.135.21 255.255.255.0
Device(config-if)# no shutdown
```

Example: Setting interface speed and duplex mode

This 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 tengigabitethernet1/1/1
Device(config-if)# speed 10
Device(config-if)# duplex full
```

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

```
Device# configure terminal
Device(config)# interface tengigabitethernet1/1/1
Device(config-if)# speed 100
```

Example: Configure the console media type

This 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
```

This 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: Configure USB inactivity timeout

This 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
```

This 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.
```