



Configuring Interface Characteristics

This chapter defines the types of interfaces on the switch and describes how to configure them.

The chapter consists of these sections:

- [Understanding Interface Types, page 8-1](#)
- [Using Interface Configuration Mode, page 8-4](#)
- [Configuring Ethernet Interfaces, page 8-8](#)
- [Configuring the System MTU, page 8-15](#)
- [Monitoring and Maintaining the Interfaces, page 8-16](#)



Note

For complete syntax and usage information for the commands used in this chapter, see the *Cisco Catalyst Blade Switch 3030 Command Reference* for this release and the online *Cisco IOS Interface Command Reference, Release 12.2*.

Understanding Interface Types

This section describes the different types of interfaces supported by the switch with references to chapters that contain more detailed information about configuring these interface types. The rest of the chapter describes configuration procedures for physical interface characteristics.

These sections describe the interface types:

- [Port-Based VLANs, page 8-1](#)
- [Switch Ports, page 8-2](#)
- [EtherChannel Port Groups, page 8-3](#)
- [Connecting Interfaces, page 8-4](#)

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. For more information about VLANs, see [Chapter 10, “Configuring VLANs.”](#) 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 normal-range VLANs (VLAN IDs 1 to 1005), use the **vlan** *vlan-id* global configuration command to enter config-vlan mode or the **vlan database** privileged EXEC command to enter VLAN database configuration mode. The VLAN configurations for VLAN IDs 1 to 1005 are saved in the VLAN database. To configure extended-range VLANs (VLAN IDs 1006 to 4094), you must use config-vlan mode with VTP mode set to transparent. Extended-range VLANs are not added to the VLAN database. When VTP mode is transparent, the VTP and VLAN configuration is saved in the switch running configuration, and you can save it in the switch startup configuration file by entering the **copy running-config startup-config** privileged EXEC command.

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

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

Switch Ports

Switch ports are Layer 2-only interfaces associated with a physical port. Switch ports belong to one or more VLANs. A switch port can be an access port or a trunk port. You can configure a port as an access port or trunk port or let the Dynamic Trunking Protocol (DTP) operate on a per-port basis to set the switchport mode by negotiating with the port on the other end of the link. Switch ports are used for managing the physical interface and associated Layer 2 protocols.

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

For detailed information about configuring access port and trunk port characteristics, see [Chapter 10, “Configuring VLANs.”](#)

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.

Two types of access ports are supported:

- Static access ports are manually assigned to a VLAN (or through a RADIUS server for use with IEEE 802.1x. For more information, see the [“Using IEEE 802.1x Authentication with VLAN Assignment”](#) section on page 7-10.)
- VLAN membership of dynamic access ports is learned through incoming packets. By default, a dynamic access port is not a member of any VLAN, and forwarding to and from the port is enabled only when the VLAN membership of the port is discovered. Dynamic access ports on the switch are assigned to a VLAN by a VLAN Membership Policy Server (VMPS). The VMPS can be a Catalyst 6500 series switch; the Cisco Catalyst Blade Switch 3030 cannot be a VMPS server.

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. For more information about voice VLAN ports, see [Chapter 12, “Configuring Voice VLAN.”](#)

Trunk Ports

A trunk port carries the traffic of multiple VLANs and by default is a member of all VLANs in the VLAN database. These trunk port types are supported:

- In an ISL trunk port, all received packets are expected to be encapsulated with an ISL header, and all transmitted packets are sent with an ISL header. Native (non-tagged) frames received from an ISL trunk port are dropped.
- 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.

For more information about trunk ports, see [Chapter 10, “Configuring VLANs.”](#)

EtherChannel Port Groups

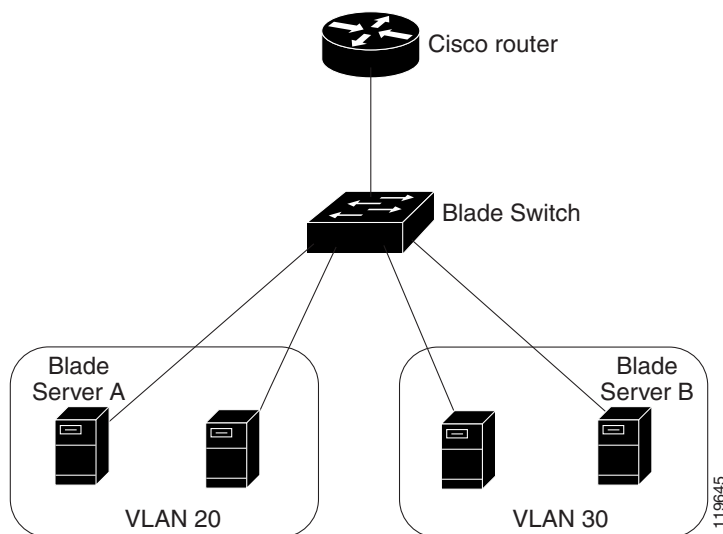
EtherChannel port groups treat multiple switch ports as one switch port. These port groups act as a single logical port for high-bandwidth connections between switches or between switches and servers. An EtherChannel balances the traffic load across the links in the channel. If a link within the EtherChannel fails, traffic previously carried over the failed link changes to the remaining links. You can group multiple trunk ports into one logical trunk port or multiple access ports into one logical access port. Most protocols operate over either single ports or aggregated switch ports and do not recognize the physical ports within the port group. Exceptions are the DTP, the Cisco Discovery Protocol (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. Use the **channel-group** interface configuration command to dynamically create the port-channel logical interface. This command binds the physical and logical ports together. For more information, see [Chapter 28, “Configuring EtherChannels and Layer 2 Trunk Failover.”](#)

Connecting Interfaces

Devices within a single VLAN can communicate directly through any switch. Ports in different VLANs cannot exchange data without going through a routing device. In the configuration shown in [Figure 8-1](#), when Blade Server A in VLAN 20 sends data to Blade Server B in VLAN 30, the data must go from Blade Server A to the switch, to the router, back to the switch, and then to Blade Server B.

Figure 8-1 Connecting VLANs with Layer 2 Switches



Using Interface Configuration Mode

The switch supports these interface types:

- Physical ports—switch ports
- VLANs—switch virtual interfaces
- Port channels—EtherChannel interfaces

You can also configure a range of interfaces (see the [“Configuring a Range of Interfaces”](#) section on [page 8-5](#)).

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

- Type—Gigabit Ethernet (`gigabitethernet` or `gi`) for 10/100/1000 Mbps Ethernet ports or small form-factor pluggable (SFP) module Gigabit Ethernet interfaces.
- Module number—The module or slot number on the switch (always 0 on the Cisco Catalyst Blade Switch 3030).
- Port number—The interface number on the switch. The port numbers always begin at 1, starting with the far left port when facing the front of the switch, for example, `gigabitethernet0/1`. For a switch with 10/100/1000 ports and SFP module ports, SFP module ports are numbered consecutively following the 10/100/1000 ports.

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

Procedures for Configuring Interfaces

These general instructions apply to all interface configuration processes.

Step 1 Enter the **configure terminal** command at the privileged EXEC prompt:

```
Switch# configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
Switch(config)#
```

Step 2 Enter the **interface** global configuration command. Identify the interface type and the number of the connector. In this example, Gigabit Ethernet port 1 is selected:

```
Switch(config)# interface gigabitethernet0/1  
Switch(config-if)#
```



Note You do not need to add a space between the interface type and interface number. For example, in the preceding line, you can specify either **gigabitethernet 0/1**, **gigabitethernet0/1**, **gi 0/1**, or **gi0/1**.

Step 3 Follow each **interface** command with the interface configuration commands that the interface requires. The commands that you enter define 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.

You can also configure a range of interfaces by using the **interface range** or **interface range macro** global configuration commands. Interfaces configured in a range must be the same type and must be configured with the same feature options.

Step 4 After you configure an interface, verify its status by using the **show** privileged EXEC commands listed in the [“Monitoring and Maintaining the Interfaces”](#) section on page 8-16.

Enter the **show interfaces** privileged EXEC command to see 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.

Configuring a Range of Interfaces

You can 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.

Beginning in privileged EXEC mode, follow these steps to configure a range of interfaces with the same parameters:

| | Command | Purpose |
|--------|---|---|
| Step 1 | configure terminal | Enter global configuration mode. |
| Step 2 | interface range { <i>port-range</i> macro <i>macro_name</i> } | Specify the range of interfaces (VLANs or physical ports) to be configured, and enter interface-range configuration mode. <ul style="list-style-type: none"> 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 the “Configuring and Using Interface Range Macros” section on page 8-7. 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. |
| Step 3 | | 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 | Return to privileged EXEC mode. |
| Step 5 | show interfaces [<i>interface-id</i>] | Verify the configuration of the interfaces in the range. |
| Step 6 | copy running-config startup-config | (Optional) Save your entries in the configuration file. |

When using the **interface range** global configuration command, note these guidelines:

- Valid entries for *port-range*:
 - vlan** *vlan-ID*, where the VLAN ID is 1 to 4094
 - gigabitethernet** *module*/{*first port*} - {*last port*}, where the module is always 0
 - port-channel** *port-channel-number* - *port-channel-number*, where the *port-channel-number* is 1 to 48



Note When you use the **interface range** command with port channels, the first and last port-channel number must be active port channels.

- You must add a space between the first interface number and the hyphen when using the **interface range** command. For example, the command **interface range gigabitethernet0/1 - 4** is a valid range; the command **interface range gigabitethernet0/1-4** is not a valid range.
- The **interface range** command only works with VLAN interfaces that have been configured with the **interface vlan** command. The **show running-config** privileged EXEC command displays the configured VLAN interfaces. VLAN interfaces not displayed by the **show running-config** command cannot be used with the **interface range** command.
- All interfaces defined in a range must be the same type (all Gigabit Ethernet ports, all EtherChannel ports, or all VLANs), but you can enter multiple ranges in a command.

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

```
Switch# configure terminal
Switch(config)# interface range gigabitethernet0/1 - 4
Switch(config-if-range)# speed 100
```

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

Configuring and Using Interface Range Macros

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.

Beginning in privileged EXEC mode, follow these steps to define an interface range macro:

| | Command | Purpose |
|--------|---|--|
| Step 1 | configure terminal | Enter global configuration mode. |
| Step 2 | define interface-range <i>macro_name</i> <i>interface-range</i> | Define the interface-range macro, and save it in NVRAM. <ul style="list-style-type: none"> 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. |
| Step 3 | interface range macro <i>macro_name</i> | Select the interface range to be configured using the values saved in the interface-range macro called <i>macro_name</i> . You can now use the normal configuration commands to apply the configuration to all interfaces in the defined macro. |
| Step 4 | end | Return to privileged EXEC mode. |
| Step 5 | show running-config include define | Show the defined interface range macro configuration. |
| Step 6 | copy running-config startup-config | (Optional) Save your entries in the configuration file. |

Use the **no define interface-range** *macro_name* global configuration command to delete a macro.

When using the **define interface-range** global configuration command, note these guidelines:

- Valid entries for *interface-range*:
 - vlan** *vlan-ID* , where the VLAN ID is 1 to 4094
 - gigabitethernet** *module*/*{first port}* - *{last port}*, where the module is always 0
 - port-channel** *port-channel-number* - *port-channel-number*, where the *port-channel-number* is 1 to .



Note

When you use the interface ranges with port channels, the first and last port-channel number must be active port channels.

- You must add a space between the first interface number and the hyphen when entering an *interface-range*. For example, **gigabitethernet0/1 - 4** is a valid range; **gigabitethernet0/1-4** is not a valid range.
- The VLAN interfaces must have been configured with the **interface vlan** command. The **show running-config** privileged EXEC command displays the configured VLAN interfaces. VLAN interfaces not displayed by the **show running-config** command cannot be used as *interface-ranges*.
- All interfaces defined as in a range must be the same type (all Gigabit Ethernet ports, all EtherChannel ports, or all VLANs), but you can combine multiple interface types in a macro.

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

```
Switch# configure terminal
Switch(config)# define interface-range enet_list gigabitethernet0/1 - 2
Switch(config)# end
Switch# show running-config | include define
define interface-range enet_list GigabitEthernet0/1 - 2
```

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

```
Switch# configure terminal
Switch(config)# define interface-range macro1 gigabitethernet0/1 - 2, gigabitethernet0/11
- 16
Switch(config)# end
```

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

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

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

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

Configuring Ethernet Interfaces

These sections contain this configuration information:

- [Default Ethernet Interface Configuration, page 8-9](#)
- [Configuring Interface Speed and Duplex Mode, page 8-10](#)
- [Configuring IEEE 802.3x Flow Control, page 8-12](#)
- [Configuring Auto-MDIX on an Interface, page 8-13](#)
- [Adding a Description for an Interface, page 8-14](#)

Default Ethernet Interface Configuration

Table 8-1 shows the Ethernet interface default configuration. For more details on the VLAN parameters listed in the table, see Chapter 10, “Configuring VLANs.” For details on controlling traffic to the port, see Chapter 19, “Configuring Port-Based Traffic Control.”

Table 8-1 Default Layer 2 Ethernet Interface Configuration

| Feature | Default Setting |
|---|---|
| Allowed VLAN range | VLANs 1– 4094. |
| Default VLAN (for access ports) | VLAN 1. |
| Native VLAN (for IEEE 802.1Q trunks) | VLAN 1. |
| VLAN trunking | Switchport mode dynamic auto (supports DTP). |
| Port enable state | All ports are enabled. |
| Port description | None defined. |
| Speed | Autonegotiate. |
| Duplex mode | Autonegotiate. |
| Flow control | Flow control is set to receive: off . It is always off for sent packets. |
| EtherChannel (PAgP) | Disabled on all Ethernet ports. See Chapter 28, “Configuring EtherChannels and Layer 2 Trunk Failover.” |
| Port blocking (unknown multicast and unknown unicast traffic) | Disabled (not blocked). See the “Configuring Port Blocking” section on page 19-6. |
| Broadcast, multicast, and unicast storm control | Disabled. See the “Default Storm Control Configuration” section on page 19-3. |
| Protected port | Disabled. See the “Configuring Protected Ports” section on page 19-5. |
| Port security | Disabled. See the “Default Port Security Configuration” section on page 19-10. |
| Port Fast | Disabled. See the “Default Optional Spanning-Tree Configuration” section on page 15-9. |
| 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. |

Configuring Interface Speed and Duplex Mode

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

Switch models include Gigabit Ethernet (10/100/1000-Mbps) ports and small form-factor pluggable (SFP) module slots supporting SFP modules.

These sections describe how to configure the interface speed and duplex mode:

- [Speed and Duplex Configuration Guidelines, page 8-10](#)
- [Setting the Interface Speed and Duplex Parameters, page 8-11](#)

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 do not support half-duplex mode.
- For SFP module ports, the speed and duplex CLI options change depending on the SFP module type:
 - The 1000BASE-*x* (where *x* is -BX, -CWDM, -LX, -SX, and -ZX) SFP module ports support the **nonegotiate** keyword in the **speed** interface configuration command. Duplex options are not supported.
 - The 1000BASE-T SFP module ports support the same speed and duplex options as the 10/100/1000-Mbps ports.
 - The 100BASE-*x* (where *x* is -BX, -FX, -FX-FE, and -LX) SFP module ports support only 100 Mbps. These modules support full- and half- duplex options but do not support autonegotiation.

For information about which SFP modules are supported on your switch, see the product release notes.

- You cannot configure duplex mode on SFP module ports; they operate in full-duplex mode except in these situations:
 - You can configure Cisco 1000BASE-T SFP modules for auto, full, or half-duplex mode.
 - Cisco 1000BASE-SX SFP modules can operate only in full-duplex mode.
- If you are connected to a device that does not support autonegotiation, you can configure speed on copper SFP ports; however, you can only configure fiber SFP ports to not negotiate (**nonegotiate**).
- If both ends of the line support autonegotiation, we highly recommend the default setting of **auto** negotiation.
- If one interface supports autonegotiation and the other end does not, configure duplex and speed on both interfaces; do not use the **auto** setting on the supported side.
- When STP is enabled and a port is reconfigured, the switch can take up to 30 seconds to check for loops. The port LED is amber while STP reconfigures.



Caution

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

Setting the Interface Speed and Duplex Parameters

Beginning in privileged EXEC mode, follow these steps to set the speed and duplex mode for a physical interface:

| | Command | Purpose |
|--------|---|--|
| Step 1 | configure terminal | Enter global configuration mode. |
| Step 2 | interface <i>interface-id</i> | Specify the physical interface to be configured, and enter interface configuration mode. |
| Step 3 | speed { 10 100 1000 auto [10 100 1000] nonegotiate } | Enter the appropriate speed parameter for the interface: <ul style="list-style-type: none"> Enter 10, 100, or 1000 to set a specific speed for the interface. The 1000 keyword is available only for 10/100/1000 Mbps ports. Enter auto to enable the interface to autonegotiate speed with the connected device. If you use the 10, 100, or the 1000 keywords with 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 Mbps but can be configured to not negotiate if connected to a device that does not support autonegotiation. For more information about speed settings, see the “Speed and Duplex Configuration Guidelines” section on page 8-10 . |
| Step 4 | duplex { auto full half } | Enter the duplex parameter for the interface. Enable half-duplex mode (for interfaces operating only at 10 or 100 Mbps). You cannot configure half-duplex mode for interfaces operating at 1000 Mbps. For more information about duplex settings, see the “Speed and Duplex Configuration Guidelines” section on page 8-10 . |
| Step 5 | end | Return to privileged EXEC mode. |
| Step 6 | show interfaces <i>interface-id</i> | Display the interface speed and duplex mode configuration. |
| Step 7 | copy running-config startup-config | (Optional) Save your entries in the configuration file. |

Use the **no speed** and **no duplex** interface configuration commands to return the interface to the default speed and duplex settings (autonegotiate). To return all interface settings to the defaults, use the **default interface** *interface-id* interface configuration command.

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

```
Switch# configure terminal
Switch(config)# interface gigabitethernet0/2
Switch(config-if)# speed 100
```

Configuring 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

Cisco Catalyst Blade Switch 3030 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 **off**.

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 *Cisco Catalyst Blade Switch 3030 Command Reference* for this release.

Beginning in privileged EXEC mode, follow these steps to configure flow control on an interface:

| | Command | Purpose |
|--------|---|--|
| Step 1 | configure terminal | Enter global configuration mode |
| Step 2 | interface <i>interface-id</i> | Specify the physical interface to be configured, and enter interface configuration mode. |
| Step 3 | flowcontrol { receive } { on off desired } | Configure the flow control mode for the port. |
| Step 4 | end | Return to privileged EXEC mode. |
| Step 5 | show interfaces <i>interface-id</i> | Verify the interface flow control settings. |
| Step 6 | copy running-config startup-config | (Optional) Save your entries in the configuration file. |

To disable flow control, use the **flowcontrol receive off** interface configuration command.

This example shows how to turn on flow control on a port:

```
Switch# configure terminal
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# flowcontrol receive on
Switch(config-if)# end
```

Configuring Auto-MDIX on an Interface

When automatic medium-dependent interface crossover (auto-MDIX) is enabled on an interface, the interface automatically detects the required cable connection type (straight through or crossover) and configures the connection appropriately. When connecting switches without the auto-MDIX feature, you must use straight-through cables to connect to devices such as servers, workstations, or routers and crossover cables to connect to other switches or repeaters. With auto-MDIX enabled, you can use either type of cable to connect to other devices, and the interface automatically corrects for any incorrect cabling. For more information about cabling requirements, see the *Cisco Catalyst Blade Switch 3030 Hardware Installation Guide*.

Auto-MDIX is enabled by default. When you enable auto-MDIX, you must also set the interface speed and duplex to **auto** so that the feature operates correctly. Auto-MDIX is supported on all 10/100/1000-Mbps interfaces and on 10/100/1000BASE-T small form-factor pluggable (SFP)-module interfaces. It is not supported on 1000BASE-SX or -LX SFP module interfaces.

Table 8-2 shows the link states that result from auto-MDIX settings and correct and incorrect cabling.

Table 8-2 Link Conditions and Auto-MDIX Settings

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

Beginning in privileged EXEC mode, follow these steps to configure auto-MDIX on an interface:

| | Command | Purpose |
|--------|---|--|
| Step 1 | configure terminal | Enter global configuration mode |
| Step 2 | interface <i>interface-id</i> | Specify the physical interface to be configured, and enter interface configuration mode. |
| Step 3 | speed auto | Configure the interface to autonegotiate speed with the connected device. |
| Step 4 | duplex auto | Configure the interface to autonegotiate duplex mode with the connected device. |
| Step 5 | mdix auto | Enable auto-MDIX on the interface. |
| Step 6 | end | Return to privileged EXEC mode. |
| Step 7 | show controllers ethernet-controller <i>interface-id</i> phy | Verify the operational state of the auto-MDIX feature on the interface. |
| Step 8 | copy running-config startup-config | (Optional) Save your entries in the configuration file. |

To disable auto-MDIX, use the **no mdix auto** interface configuration command.

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

```
Switch# configure terminal
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# speed auto
Switch(config-if)# duplex auto
Switch(config-if)# mdix auto
Switch(config-if)# end
```

Adding a Description for an Interface

You can add a description about an interface to help you remember its function. The description appears in the output of these privileged EXEC commands: **show configuration**, **show running-config**, and **show interfaces**.

Beginning in privileged EXEC mode, follow these steps to add a description for an interface:

| | Command | Purpose |
|--------|---|---|
| Step 1 | configure terminal | Enter global configuration mode. |
| Step 2 | interface <i>interface-id</i> | Specify the interface for which you are adding a description, and enter interface configuration mode. |
| Step 3 | description <i>string</i> | Add a description (up to 240 characters) for an interface. |
| Step 4 | end | Return to privileged EXEC mode. |
| Step 5 | show interfaces <i>interface-id</i> description or show running-config | Verify your entry. |
| Step 6 | copy running-config startup-config | (Optional) Save your entries in the configuration file. |

Use the **no description** interface configuration command to delete the description.

This example shows how to add a description on a port and how to verify the description:

```
Switch# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface gigabitethernet0/2
Switch(config-if)# description Connects to Marketing
Switch(config-if)# end
Switch# show interfaces gigabitethernet0/2 description
Interface Status      Protocol Description
Gi0/2    admin down    down    Connects to Marketing
```

Configuring the System MTU

The default maximum transmission unit (MTU) size for frames received and transmitted on all interfaces on the switch is 1500 bytes. You can increase the MTU size for all interfaces operating at 10 or 100 Mbps by using the **system mtu** global configuration command. You can increase the MTU size to support jumbo frames on all Gigabit Ethernet interfaces by using the **system mtu jumbo** global configuration command.

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

You cannot set the MTU size for an individual interface; you set it for all 10/100 or all Gigabit Ethernet interfaces on the switch. When you change the system or jumbo MTU size, you must reset the switch before the new configuration takes effect.

Frames sizes that can be received by the switch CPU are limited to 1998 bytes, no matter what value was entered with the **system mtu** or **system mtu jumbo** commands. Although frames that are forwarded are typically not received by the CPU, in some cases packets are sent to the CPU, such as traffic sent to control traffic, SNMP, or Telnet.



Note

If Gigabit Ethernet interfaces are configured to accept frames greater than the 10/100 interfaces, jumbo frames received on a Gigabit Ethernet interface and sent on a 10/100 interface are dropped.

Beginning in privileged EXEC mode, follow these steps to change MTU size for all 10/100 or Gigabit Ethernet interfaces:

| | Command | Purpose |
|--------|---|---|
| Step 1 | configure terminal | Enter global configuration mode. |
| Step 2 | system mtu bytes | (Optional) Change the MTU size for all interfaces on the switch that are operating at 10 or 100 Mbps. The range is 1500 to 1998 bytes; the default is 1500 bytes. |
| Step 3 | system mtu jumbo bytes | (Optional) Change the MTU size for all Gigabit Ethernet interfaces on the switch. The range is 1500 to 9000 bytes; the default is 1500 bytes. |
| Step 4 | end | Return to privileged EXEC mode. |
| Step 5 | copy running-config startup-config | Save your entries in the configuration file. |
| Step 6 | reload | Reload the operating system. |

If you enter a value that is outside the allowed range for the specific type of interface, the value is not accepted.

Once the switch reloads, you can verify your settings by entering the **show system mtu** privileged EXEC command.

This example shows how to set the maximum packet size for a Gigabit Ethernet port to 1800 bytes:

```
Switch(config)# system jumbo mtu 1800
Switch(config)# exit
Switch# reload
```

This example shows the response when you try to set Gigabit Ethernet interfaces to an out-of-range number:

```
Switch(config)# system mtu jumbo 25000
                        ^
% Invalid input detected at '^' marker.
```

Monitoring and Maintaining the Interfaces

These sections contain interface monitoring and maintenance information:

- [Monitoring Interface Status, page 8-16](#)
- [Clearing and Resetting Interfaces and Counters, page 8-17](#)
- [Shutting Down and Restarting the Interface, page 8-17](#)

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 8-3](#) lists some of these interface monitoring commands. (You can display the full list of **show** commands by using the **show ?** command at the privileged EXEC prompt.) These commands are fully described in the *Cisco IOS Interface Command Reference, Release 12.2*.

Table 8-3 Show Commands for Interfaces

| Command | Purpose |
|---|--|
| show interfaces [<i>interface-id</i>] | Display the status and configuration of all interfaces or a specific interface. |
| show interfaces <i>interface-id</i> status [err-disabled] | Display interface status or a list of interfaces in an error-disabled state. |
| show interfaces [<i>interface-id</i>] switchport | Display administrative and operational status of switching ports. |
| show interfaces [<i>interface-id</i>] description | Display the description configured on an interface or all interfaces and the interface status. |
| show ip interface [<i>interface-id</i>] | Display the usability status of all interfaces configured for IP routing or the specified interface. |
| show interface [<i>interface-id</i>] stats | Display the input and output packets by the switching path for the interface. |
| show interfaces transceiver properties | (Optional) Display speed and duplex settings on the interface. |
| show interfaces [<i>interface-id</i>] [{ transceiver properties detail }] [<i>module number</i>] | Display physical and operational status about an SFP module. |
| show running-config interface [<i>interface-id</i>] | Display the running configuration in RAM for the interface. |
| show version | Display the hardware configuration, software version, the names and sources of configuration files, and the boot images. |
| show controllers ethernet-controller <i>interface-id</i> phy | Display the operational state of the auto-MDIX feature on the interface. |

Clearing and Resetting Interfaces and Counters

Table 8-4 lists the privileged EXEC mode **clear** commands that you can use to clear counters and reset interfaces.

Table 8-4 Clear Commands for Interfaces

| Command | Purpose |
|---|--|
| clear counters [<i>interface-id</i>] | Clear interface counters. |
| clear interface <i>interface-id</i> | Reset the hardware logic on an interface. |
| clear line [<i>number</i> console 0 <i>vtty number</i>] | Reset the hardware logic on an asynchronous serial line. |

To clear the interface counters shown by the **show interfaces** privileged EXEC command, use the **clear counters** privileged EXEC command. The **clear counters** command clears all current interface counters from the interface unless you specify optional arguments that clear only a specific interface type from a specific interface number.



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.

Shutting Down and Restarting the Interface

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

Beginning in privileged EXEC mode, follow these steps to shut down an interface:

| | Command | Purpose |
|--------|---|--|
| Step 1 | configure terminal | Enter global configuration mode. |
| Step 2 | interface { <i>vlan vlan-id</i> } {{ <i>fastethernet</i> <i>gigabitethernet</i> } <i>interface-id</i> } { port-channel <i>port-channel-number</i> } | Select the interface to be configured. |
| Step 3 | shutdown | Shut down an interface. |
| Step 4 | end | Return to privileged EXEC mode. |
| Step 5 | show running-config | Verify your entry. |

Use the **no shutdown** interface configuration command to restart the interface.

To verify that an interface is disabled, enter the **show interfaces** privileged EXEC command. A disabled interface is shown as *administratively down* in the display.

