CHAPTER 4

Configuring Ethernet, Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet Switching

This chapter describes how to use the command-line interface (CLI) to configure Ethernet, Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet switching on the Catalyst 6500 series switches. The configuration tasks in this chapter apply to Ethernet, Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet switching modules, as well as to the uplink ports on the supervisor engine.

Note For complete syntax and usage information for the commands that are used in this chapter, refer to the *Catalyst 6500 Series Switch Command Reference* publication.

This chapter consists of these sections:

Understanding How Ethernet Works

Catalyst 6500 series switches support simultaneous, parallel connections between Ethernet segments. Switched connections between Ethernet segments last only for the duration of the packet. New connections can be made between different segments for the next packet.

Catalyst 6500 series switches solve congestion problems that are caused by high-bandwidth devices and a large number of users by assigning each device (for example, a server) to its own 10-, 100-, 1000-, or 10000-Mbps segment. Because each Ethernet port on the switch represents a separate Ethernet segment, the servers that are in a properly configured switched environment achieve full access to the bandwidth.

Because collisions are a major bottleneck in the Ethernet networks, an effective solution is full-duplex communication, which is an option for any 10- or 100-Mbps port on a Catalyst 6500 series switch (the Gigabit Ethernet and 10-Gigabit Ethernet ports are always full duplex). Typically, Ethernet operates in half-duplex mode, which means that stations can either receive or transmit. In full-duplex mode, two stations can transmit and receive at the same time. When packets can flow in both directions simultaneously, effective Ethernet bandwidth doubles to 20 Mbps for 10-Mbps ports and to 200 Mbps for Fast Ethernet ports. The Gigabit Ethernet and 10-Gigabit Ethernet ports on the Catalyst 6500 series switches are full duplex only (2-Gbps and 20-Gbps effective bandwidth, respectively).

These sections describe Ethernet:

Switching Frames Between Segments

Each Ethernet port on a Catalyst 6500 series switch can connect to a single workstation or server or to a hub through which workstations or servers connect to the network.

Ports on a typical Ethernet hub all connect to a common backplane within the hub, and the bandwidth of the network is shared by all devices that are attached to the hub. If two stations establish a session that uses a significant level of bandwidth, the network performance of all other stations that are attached to the hub is degraded.

To reduce degradation, the switch treats each port as an individual segment. When the stations on different ports need to communicate, the switch forwards the frames from one port to the other port at wire speed to ensure that each session receives full bandwidth.

To switch frames between ports efficiently, the switch maintains an address table. When a frame enters the switch, it associates the MAC address of the sending station with the port on which it was received.

Building the Address Table

Catalyst 6500 series switches build the address table by using the source address of the received frames. When the switch receives a frame for a destination address that is not listed in its address table, it floods the frame to all ports of the same VLAN except for the port that received the frame. When the destination station replies, the switch adds its relevant source address and port ID to the address table. The switch then forwards subsequent frames to a single port without flooding to all ports.

The address table can store at least 32,000 address entries without flooding any entries. The switch uses an aging mechanism, which is defined by a configurable aging timer, so if an address remains inactive for a specified number of seconds, it is removed from the address table.

Understanding Port Negotiation

Note The **set port negotiation** command is supported on Gigabit Ethernet ports only; it is not supported on the WS-X6316-GE-TX and WS-X6516-GE-TX modules. If a port does not support this command, this message appears: “Feature not supported on Port N/N,” where N/N is the module and the port number.

Note You cannot configure port negotiation on 1000BASE-TX (copper) Gigabit Ethernet ports in this release. If you insert a 1000BASE-TX GBIC in the port that was previously configured as negotiation disabled, the negotiation-disabled setting is ignored and the port operates in negotiation-enabled mode.

Note Port negotiation does not involve negotiating port speed. You cannot disable port negotiation with the **set port speed** command.

Port negotiation exchanges flow-control parameters, remote fault information, and duplex information. Configure port negotiation with the **set port negotiation** command. Port negotiation is enabled by default.

Note When you enable port negotiation on the 16-port 10/100/1000BASE-T Ethernet modules, the system autonegotiates flow control only.

The ports on both ends of a link must have the same setting. The link will not come up if the ports at each end of the link are set inconsistently (port negotiation is enabled on one port and is disabled on the other port).

Table 4-1 shows the four possible port negotiation configurations and the resulting link status for each configuration.

Table 4-1 Port Negotiation Configuration and Possible Link Status

Port Negotiation State		Link Status	
Near End[1]	Far End[2]	Near End	Far End
Off	Off	Up	Up
On	On	Up	Up
Off	On	Up	Down
On	Off	Down	Up

1. Near End refers to the local port.
2. Far End refers to the port at the other end of the link.

Default Ethernet, Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet Configuration

Table 4-2 shows the Ethernet, Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet default configuration.

Table 4-2 Ethernet Default Configuration

Feature	Default Value
Port enable state	All ports are enabled
Port name	None

Table 4-2 Ethernet Default Configuration (continued)

Feature	Default Value
Duplex mode	• Half duplex for 10-Mbps Ethernet ports • Autonegotiate speed and duplex for 10/100-Mbps Fast Ethernet ports • Autonegotiate duplex for 100-Mbps Fast Ethernet ports • Full duplex for 1000-Mbps Gigabit Ethernet ports • Full duplex for 10000-Mbps Gigabit Ethernet ports
Flow control (10-Gigabit Ethernet)	Flow control set to on for receive (Rx) and off for transmit (Tx)[1]
Flow control (Gigabit Ethernet)	Flow control set to off for receive (Rx) and desired for transmit (Tx)
Flow control (other Ethernet)	Flow control set to off for receive (Rx); transmit (Tx) not supported
Spanning Tree Protocol (STP)	Enabled for VLAN 1
Native VLAN	VLAN 1
Port VLAN cost	• 100 for 10-Mbps Ethernet ports • 19 for 10/100-Mbps Fast Ethernet ports • 19 for 100-Mbps Fast Ethernet ports • 4 for 1000-Mbps Gigabit Ethernet ports • 1 for 10000-Mbps Gigabit Ethernet ports
EtherChannel	Disabled on all Ethernet ports
Jumbo frames	Disabled on all Ethernet ports

1. On WS-X6502-10-Gigabit Ethernet ports, flow control on the receive side is always on and cannot be set to off. On WS-X6704, WS-X6708-10-Gigabit Ethernet ports and Supervisor Engine 720-10 Gigabit Ethernet uplink ports flowcontrol for both send and receive side can be set to off.

Setting the Port Configuration

These sections describe how to configure Ethernet, Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet switching on the Catalyst 6500 series switches:

Configuring Supervisor Engine 720 Ports

Supervisor Engine 720, port 1 has a small form-factor pluggable (SFP) connector and no unique configuration options.

Note Cisco WS-X6408A-GBIC, which is an 8-port Gigabit Ethernet interface module for the Catalyst 6500 Series switches, is supported on Supervisor Engine 720.

Supervisor Engine 720, port 2 has an RJ-45 connector and an SFP connector (default). To use the RJ-45 connector, you must change the configuration.

To configure port 2 on Supervisor Engine 720, perform this task in privileged mode:

Task	Command
Configure port 2 on Supervisor Engine 720.	**set port media-type** *mod/port* {**rj45** | **sfp**}

This example shows how to configure port 2 on Supervisor Engine 720 to use the RJ-45 connector:

```
Console> (enable) set port media-type 5/2 rj45
Port 5/2 media type set to RJ-45.
Console> (enable)
```

Setting the Port Name

You can set the port names on Ethernet, Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet switching modules to facilitate switch administration.

To set the port name, perform this task in privileged mode:

	Task	Command
Step 1	Set a port name.	**set port name** *mod/port* [*name_string*]
Step 2	Verify that the port name is configured.	**show port** [*mod*[*/port*]]

This example shows how to set the name for ports 1/1 and 1/2 and verify that the port names are configured correctly:

```
Console> (enable) set port name 1/1 Router Connection
Port 1/1 name set.
Console> (enable) set port name 1/2 Server Link
Port 1/2 name set.
```

```
Console> (enable) show port 1
Port  Name               Status     Vlan         Duplex Speed Type
----- ------------------ ---------- ----------   ------ ----- ------------
 1/1  Router Connection  connected  trunk          full  1000 1000BaseSX
 1/2  Server Link        connected  trunk          full  1000 1000BaseSX
.
.
.
Last-Time-Cleared
--------------------------
Wed Jun 16 1999, 16:25:57
Console> (enable)
```

Setting the Port Speed

You can configure the port speed on 10/100-Mbps Ethernet switching modules. Use the **auto** keyword to autonegotiate the port's speed and duplex mode with the neighboring port.

Note If the port speed is set to **auto** on a 10/100-Mbps Ethernet port, both speed and duplex are autonegotiated.

Use the **auto-10-100** keyword on ports that support speeds of 10/100/1000 Mbps. Using the **auto-10-100** keyword makes the port behave the same as a 10/100-Mbps port that has the speed set to **auto**. The speed and duplex are negotiated (the 1000-Mbps speed does not take part in the negotiation).

To set the port speed of an Ethernet port, perform this task in privileged mode:

	Task	Command
Step 1	Set the port speed of an Ethernet port.	**set port speed** *mod/port* {**10** | **100** | **1000** | **auto** | **auto-10-100**}
Step 2	Verify that the speed of the port is configured correctly.	**show port** [*mod*[*/port*]]

This example shows how to set the port speed to 100 Mbps on port 2/2:

```
Console> (enable) set port speed 2/2 100
Port 2/2 speed set to 100 Mbps.
Console> (enable)
```

This example shows how to make port 2/1 autonegotiate speed and duplex with the neighboring port:

```
Console> (enable) set port speed 2/1 auto
Port 2/1 speed set to auto-sensing mode.
Console> (enable)
```

Setting the Port Duplex Mode

You can set the port duplex mode to full or half duplex for Ethernet and Fast Ethernet ports.

Note Gigabit Ethernet and 10-Gigabit Ethernet are full duplex only. You cannot change the duplex mode on Gigabit Ethernet and 10-Gigabit Ethernet ports.

Note If the port speed is set to **auto** on a 10/100-Mbps Ethernet port, both speed and duplex are autonegotiated. You cannot change the duplex mode of autonegotiation ports.

To set the duplex mode of a port, perform this task in privileged mode:

	Task	Command
Step 1	Set the duplex mode of a port.	**set port duplex** *mod/port* {**full** | **half**}
Step 2	Verify that the duplex mode of the port is configured correctly.	**show port** [*mod*[*/port*]]

This example shows how to set the duplex mode to half duplex on port 2/1:

```
Console> (enable) set port duplex 2/1 half
Port 2/1 set to half-duplex.
Console> (enable)
```

Enabling or Disabling Auto-MDI/MDIX

With auto-MDI/MDIX you can use either a straight or crossover cable, and the module will automatically detect and adjust for the cable type. Auto-MDI/MDIX works with the speed set to auto/1000 Mbps but not with the speed set to 10 Mbps or 100 Mbps. The link will come up with either a straight or crossover cable if the speed is set to auto/1000 using the **set port speed** *mod/port* **auto** command or the **set port speed** *mod/port* **1000** command. The link comes up even if the speed is autonegotiated at 10 Mbps or 100 Mbps in **auto** mode. However, if you enter the **set port speed** *mod/port* **10** command or the **set port speed** *mod/port* **100** command, the link fails to come up if the wrong cable is used.

Auto-MDI/MDIX has always been enabled on the following modules:

- WS-X6548-RJ-45, WS-X6548-RJ-21, WS-X6148-GE-TX, WS-X6548-GE-TX

  Auto-MDI/MDIX works in 10-, 100-, and 1000-Mbps modes with autonegotiated and fixed speeds.
- WS-X6516-GE-TX

  Auto-MDI/MDIX works with the speed set to auto/1000 Mbps but not with the speed set to 10 Mbps or 100 Mbps.
- WS-X6316-GE-TX

With software release 8.2(1) and later releases, auto-MDIX is also enabled on the following modules:

- WS-X6748-GE-TX, Supervisor Engine 720 port 2 (RJ-45)

  Auto-MDI/MDIX works with the speed set to auto/1000 but not with the speed set to 10 Mbps or 100 Mbps
- WS-X6148X2-RJ-45, WS-X6148X2-45AF

  Auto-MDI/MDIX works with the speed set to auto but not with the speed set to 10 Mbps or 100 Mbps.

Note Auto-MDI/MDIX is not supported on any other 10/100-Mbps Ethernet modules or GBIC, SFP, and XENPAK ports.

With software release 8.3(1) and later releases, the **set port auto-mdix** *mod/port* {**enable** | **disable**} command is introduced to disable auto-MDI/MDIX on all the modules that currently have this feature enabled by default. Use the **show port auto-mdix** [*mod*[/*port*]] command to display auto-MDI/MDIX settings.

Configuring IEEE 802.3x Flow Control

Gigabit Ethernet and 10-Gigabit Ethernet ports on the Catalyst 6500 series switches use flow control to inhibit the transmission of packets to the port for a period of time; other Ethernet ports use flow control to respond to flow-control requests.

If a Gigabit Ethernet or a 10-Gigabit Ethernet port receive buffer becomes full, the port transmits a "pause" packet that tells remote ports to delay sending more packets for a specified period of time. All Ethernet ports (10000 Mbps, 1000 Mbps, 100 Mbps, and 10 Mbps) can receive and act upon "pause" packets from other devices.

Enter the **set port flow control** command to configure flow control on ports. Table 4-3 lists the **set port flowcontrol** command keywords and describes their functions.

Table 4-3 Ethernet-Flow Control Keyword Functions

Keywords	Function
receive on[1]	The port uses flow control dictated by the neighboring port.
receive desired	The port uses flow control if the neighboring port uses it and does not use flow control if the neighboring port does not use it.
receive off	The port does not use flow control, regardless of whether flow control is requested by the neighboring port.
send on[2]	The port sends flow-control frames to the neighboring port.
send desired[2]	The port sends flow-control frames to the port if the neighboring port asks to use flow control.
send off[2]	The port does not send flow-control frames to the neighboring port.

1. On WS-X6502-10-Gigabit Ethernet ports, flow control on the receive side is always on and cannot be set to off. On WS-X6704, WS-X6708-10-Gigabit Ethernet ports and Supervisor Engine 720-10 Gigabit Ethernet uplink ports flowcontrol for both send and receive side can be set to off.
2. Supported only on Gigabit Ethernet and 10-Gigabit Ethernet ports.

To configure flow control, perform this task in privileged mode:

	Task	Command
Step 1	Set the flow-control parameters.	**set port flowcontrol** *mod/port* {**receive** | **send**} {**off** | **on** | **desired**}
Step 2	Verify the flow-control configuration.	**show port flowcontrol**

This example shows how to turn on transmit and receive flow control and verify the flow-control configuration:

```
Console> (enable) set port flowcontrol 3/1 send on
Port 3/1 will send flowcontrol to far end.
```

```
Console> (enable) set port flowcontrol 3/1 receive on
Port 3/1 will require far end to send flow control
Console> (enable) show port flowcontrol
Port  Send-Flowcontrol  Receive-Flowcntl  RxPause  TxPause
      Admin   Oper      Admin   Oper
----- ----------------  ----------------  -------  -------
 3/1  on      disagree  on      disagree  0        0
 3/2  off     off       off     off       0        0
 3/3  desired on        desired off       10       10
Console> (enable)
```

Enabling and Disabling Port Negotiation

To enable port negotiation, perform this task in privileged mode:

	Task	Command
Step 1	Enable port negotiation.	**set port negotiation** *mod*/*port* **enable**
Step 2	Verify the port negotiation configuration.	**show port negotiation** [*mod*/*port*]

This example shows how to enable port negotiation and verify the configuration:

```
Console> (enable) set port negotiation 2/1 enable
Port 2/1 negotiation enabled
Console> (enable) show port negotiation 2/1
Port   Link Negotiation
-----  ----------------
 2/1   enabled
Console> (enable)
```

To disable port negotiation, perform this task in privileged mode:

	Task	Command
Step 1	Disable port negotiation.	**set port negotiation** *mod*/*port* **disable**
Step 2	Verify the port negotiation configuration.	**show port negotiation** [*mod*/*port*]

This example shows how to disable port negotiation and verify the configuration:

```
Console> (enable) set port negotiation 2/1 disable
Port 2/1 negotiation disabled
Console> (enable) show port negotiation 2/1
Port   Link Negotiation
-----  ----------------
 2/1   disabled
Console> (enable)
```

Changing the Default Port Enable State

Note Changing the default port enable state applies to all port types, not just Ethernet.

When you enter the **clear config all** command or in the event of a configuration loss, all ports collapse into VLAN 1. This situation might cause a security and network instability problem. Entering the **set default portstatus** command puts all ports into a disable state and blocks the traffic flowing through the ports during a configuration loss. You can then manually configure the ports back to the enable state.

The default port status configuration is stored on the chassis. The configuration is tied to a chassis and not to the supervisor engine. The **clear config all** command uses this setting to determine whether ports should be enabled or disabled when returning to default configuration. The **clear config all** command does not change the default port status setting on the chassis. The output of the **show config** command shows the current default port status configuration.

To change the port enable state, perform this task in privileged mode:

	Task	Command
Step 1	Change the port enable state.	**set default portstatus** {**enable** | **disable**}
Step 2	Display the port enable state.	**show default**

This example shows how to change the default port enable state from enabled to disabled:

```
Console> (enable) set default portstatus disable
Default port status set to disable.
Console> (enable)
```

This example shows how to display the port enable state:

```
Console> (enable) show default
portstatus: disable
Console> (enable)
```

Setting the Port Debounce Timer

You can set the port debounce timer on a per-port basis for Ethernet, Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet ports. When you set the port debounce timer, the switch delays notifying the main processor of a link change that can decrease traffic loss due to network reconfiguration.

Caution Enabling the port debounce timer causes link up and link down detections to be delayed, resulting in loss of data traffic during the debouncing period. This situation might affect the convergence and reconvergence of various Layer 2 and Layer 3 protocols.

Table 4-4 lists the time delay that occurs before the switch notifies the main processor of a link change before and after the switch enables the debounce timer.

Table 4-4 Port Debounce Timer Delay Time

Port Type	Debounce Timer Disabled	Debounce Timer Enabled
10BASE-FL ports	300 milliseconds	3100 milliseconds
10/100BASE-TX ports	300 milliseconds	3100 milliseconds
100BASE-FX ports	300 milliseconds	3100 milliseconds
10/100/1000BASE-TX ports	300 milliseconds	3100 milliseconds
1000BASE-TX ports	300 milliseconds	3100 milliseconds

Table 4-4 Port Debounce Timer Delay Time (continued)

Port Type	Debounce Timer Disabled	Debounce Timer Enabled
Fiber Gigabit Ethernet ports	10 milliseconds	100 milliseconds
10-Gigabit Ethernet ports	10 milliseconds	100 milliseconds

To set the debounce timer on a port, perform this task in privileged mode:

	Task	Command
Step 1	Enable the debounce timer for a port.	**set port debounce** *mod num/port num* {**enable** | **disable**}
Step 2	Verify that the debounce timer of the port is configured correctly.	**show port debounce** [*mod* | *mod_num/port_num*]

This example shows how to enable the debounce timer on port 2/1:

```
Console> (enable) set port debounce 2/1 enable
Debounce is enabled on port 2/1
Warning: Enabling port debounce causes Link Up/Down detections to be delayed.
It results in loss of data traffic during debouncing period, which might
affect the convergence/reconvergence of various Layer 2 and Layer 3 protocols.
Use with caution.
Console> (enable)
```

This example shows how to display the per-port debounce timer settings:

```
Console> (enable) show port debounce
Port   Debounce link timer
-----  ---------------
 2/1   enable
 2/2   disable
Console> (enable)
```

Modifying the Port Debounce Timer Setting

Note Modifying the port debounce timer setting is possible only on fiber Gigabit Ethernet ports.

You can increase the port debounce timer value in increments of 100 up to 5000 milliseconds. You do not need to enable the debounce timer on the port before adjusting the timer value. Specifying any timer value that is greater than the default value in the disabled state enables the debounce timer.

To modify the port debounce timer setting on a port, perform this task in privileged mode:

	Task	Command
Step 1	Modify the port debounce timer setting.	**set port debounce** *mod num/port num* **delay** *time*
Step 2	Verify that the port debounce timer setting has been modified.	**show port debounce** [*mod* | *mod_num/port_num*]

This example shows how to modify the port debounce timer setting on port 2/1:

```
Console> (enable) set port debounce 2/1 delay 500
Debounce time for port 2/1 set to 500 ms.
Warning:Enabling port debounce causes Link Up/Down detections to be delayed.
It results in loss of data traffic during debouncing period, which might
affect the convergence/reconvergence of various Layer 2 and Layer 3 protocols.
Use with caution.
Console> (enable)
```

This example shows how to display the per-port debounce timer setting on port 2/1:

```
Console> (enable) show port debounce 2/1
Port   Debounce link timer
-----  ------------------
 2/1   enabled (500 ms)
Console> (enable)
```

Configuring a Timeout Period for Ports in errdisable State

A port is in errdisable state if it is enabled in NVRAM but is disabled at runtime by any process. For example, if UniDirectional Link Detection (UDLD) detects a unidirectional link, the port shuts down at runtime. However, because the NVRAM configuration for the port is enabled (you have not disabled the port), the port status is shown as errdisable.

If a port goes into errdisable state, it is reenabled automatically after a selected time interval. With the timeout enhancement, you can manually prevent a port from being enabled by setting the errdisable timeout for that port to disable using the **set port errdisable-timeout** *mod/port* **disable** command.

A global timer is maintained for all ports. At every *t* seconds, where *t* is the user-configurable timeout, a process checks to see if any ports are in errdisable state. If there are ports in errdisable state, only those ports that have errdisable timeout set (enabled) are reenabled through SCP messages.

By default, all errdisabled ports are reenabled when the global timer times out.

A port enters errdisable state for the following reasons (these reasons appear as configuration options within the **set errdisable-timeout enable** command):

- ARP inspection
- Broadcast suppression
- BPDU port-guard
- CAM monitor
- Channel misconfiguration
- Crossbar failure
- Duplex mismatch
- Layer 2 protocol tunnel misconfiguration
- Layer 2 protocol tunnel threshold exceeded
- Layer 2 protocol tunnel CDP threshold exceeded
- Layer 2 protocol tunnel STP threshold exceeded
- Layer 2 protocol tunnel VTP threshold exceeded
- Link errors RX threshold exceeded
- Link errors TX threshold exceeded

- UDLD
- Other (reasons other than the above)
- All (apply errdisable timeout for all of the above reasons)

You can enable or disable errdisable timeout for each of the above listed reasons. If you specify "other," all ports that are errdisabled by causes *other* than the reasons listed are enabled for errdisable timeout. If you specify "all," all ports that are errdisabled for any reason are enabled for errdisable timeout.

The errdisable feature is disabled by default. The default interval for enabling a port is 300 seconds. The allowable interval range is 30 to 86400 seconds (30 seconds to 24 hours).

To enable and configure the timeout period for ports in the errdisable state, perform these tasks in privileged mode:

Task	Command
Prevent a port from being reenabled at timeout after it goes into the errdisable state.	**set port errdisable-timeout** *mod/port* **disable**
Enable errdisable timeout for the BPDU guard causes.	**set errdisable-timeout enable bpdu-guard**
Enable errdisable timeout for all causes.	**set errdisable-timeout enable all**
Set the errdisable timeout interval.	**set errdisable-timeout interval** *interval*
Display the errdisable timeout configuration.	**show errdisable-timeout**

This example shows how to prevent port 3/3 from being reenabled at timeout after it goes into the errdisable state:

```
Console> (enable) set port errdisable-timeout 3/3 disable
Successfully disabled errdisable-timeout for port 3/3.
Console> (enable)
```

This example shows how to enable errdisable timeout for BPDU guard causes:

```
Console> (enable) set errdisable-timeout enable bpdu-guard
Successfully enabled errdisable-timeout for bpdu-guard.
Console> (enable)
```

This example shows how to enable errdisable timeout for all causes:

```
Console> (enable) set errdisable-timeout enable all
Successfully enabled errdisable-timeout for all.
Console> (enable)
```

This example shows how to set the errdisable timeout interval to 450 seconds:

```
Console> (enable) set errdisable-timeout interval 450
Successfully set errdisable timeout to 450 seconds.
Console> (enable)
```

This example shows how to display the errdisable timeout configuration:

```
Console> (enable) show errdisable-timeout
ErrDisable Reason                     Timeout Status
------------------------------------- --------------
arp-inspection                        enable
bcast-suppression                     enable
bpdu-guard                            enable
cam-monitor                           enable
channel-misconfig                     enable
crossbar-fallback                     enable
duplex-mismatch                       enable
gl2pt-ingress-loop                    enable
gl2pt-threshold-exceed                enable
gl2pt-cdp-threshold-exceed            enable
gl2pt-stp-threshold-exceed            enable
gl2pt-vtp-threshold-exceed            enable
link-rxcrc                            enable
link-txcrc                            enable
udld                                  enable
other                                 enable

Interval: 450 seconds

Port  ErrDisable Reason    Port ErrDisableTimeout  Action on Timeout
----  -------------------  ----------------------  -----------------
Console> (enable)
```

Configuring Automatic Module Shutdown

When you enable automatic module shutdown, you can manage your network connectivity issues. A module that frequently resets itself can disrupt traffic load balancing. By enabling the automatic module shutdown, you can disable a module that continually resets due to any hardware or software problems and limit the number of times that the module resets itself before shutting down completely.

You can also shut down a module manually using the **set module disable** or the **set module power down** commands.

After the module shuts down, you must reenable the module manually.

By default, automatic module shutdown is disabled. When you enable automatic module shutdown, the default is that the module can reset itself three times within two minutes.

You must configure these two parameters before an automatic shutdown can occur:

- Frequency—Allows you to specify the threshold value for an automatic module shutdown. When the number of resets reaches the value that is assigned to this option, the Ethernet module can perform an automatic shutdown.
- Period—Allows you to specify the time period in which the number of resets must occur. The period is measured from one of these conditions:
  - When the switch first comes up
  - When the supervisor engine performs a switchover
  - When the Ethernet module is powered up
  - When the module's autoshut counters are cleared

When the frequency threshold is reached and occurs within the defined period, the Ethernet module automatically shuts down and this sample syslog message is displayed:

```
%SYS-5-MOD_AUTOSHUT: Module 2 shutdown automatically, reset 4 times in last 5
minutes due to inband failure
```

When the frequency threshold is reached and occurs outside the defined period, the module does not automatically shut down and this sample syslog message is displayed:

```
%%SYS-4-MOD_AUTOSHUT_SLOW:Module 1 reset frequency exceeded threshold but over 46
mins. Hence NOT powering down module
```

The run-time variable states for the Ethernet module do not synchronize with the standby supervisor engine. The output of the **show autoshut** command on a standby supervisor engine does not track with the number of resets or the reasons for the resets. If the module is powered down by the **autoshut** command, the output stays the same.

You do not have to enable an automatic module shutdown in order to track the number of resets. You can track resets even if you do not enable an automatic module shutdown.

The runtime counters are cleared only for these conditions:

- When you enter the **clear autoshut** command
- When the switch resets
- At module power up
- At supervisor engine switchover

Note An automatic module shutdown is supported on Ethernet modules only.

To enable and configure an automatic module shutdown, perform one of these tasks in privileged mode:

Task	Command
Enable an automatic module shutdown on a module.	**set module autoshut enable** *mod num*
Disable an automatic module shutdown on a module.	**set module autoshut disable** *mod num*
Set the threshold of the number of times that the module can reset itself.	**set autoshut frequency** *num*
Set the period (in minutes) over which the frequency is valid.	**set autoshut period** *minutes*
Clear the run-time counters on a specific module.	**clear autoshut counters** *mod num*
Reset the autoshut frequency to the default setting.	**clear autoshut frequency**
Reset the autoshut period to the default setting.	**clear autoshut period**
Display the automatic module shutdown configuration and current status information.	**show autoshut**

This example shows how to enable an automatic module shutdown on a module:

```
Console> (enable) set module autoshut enable 2
```

This example shows how to disable an automatic module shutdown on a module:

```
Console> (enable) set module autoshut disable 2
```

This example shows how to set the threshold of the number of times that the module can reset itself:

```
Console> (enable) set autoshut frequency 4
```

This example shows how to set the period (in minutes) over which the frequency is valid:

```
Console> (enable) set autoshut period 3
```

This example shows how to clear the run-time counters on a specific module:

```
Console> (enable) clear autoshut counters 3
Automatic shutdown counters cleared for module 3
Console> (enable)
```

This example shows how to reset the automatic module shutdown frequency to the default setting:

```
Console> (enable) clear autoshut frequency
```

This example shows how to reset the automatic module shutdown period to the default setting:

```
Console> (enable) clear autoshut period
```

This example shows how to display the automatic module shutdown configuration and current status information:

```
Console> (enable) show autoshut
AutoShut Frequency:     3 times
AutoShut Period:        5 minutes

Mod Autoshut Current  Number Reason for last Time of last reset
num status   status   resets reset
--- -------- -------- ------ ---------------- -------------------------
1   NA       ok       -      -                -
2   enabled  shutdown 4       inband failure  Mon Jul 14 2003, 22:55:45
3   disabled ok       0      None             -
4   enabled  ok       1      scp failure      Mon Jul 14 2003, 21:03:17
Console> (enable)
```

Configuring Port Error Detection

Note All ports in an EtherChannel should have the same port error detection settings.

To enable or disable port error detection on a port, perform this task in privileged mode (the default port setting for inerrors, RXCRC, and TXCRC is disabled):

	Task	Command
Step 1	Enable or disable port error detection on a port.	**set port errordetection** *mod/port* {**inerrors** | **rxcrc** | **txcrc**} {**disable** | **enable**}
Step 2	Verify the port configuration.	**show port errordetection** [*mod* | *mod/port*]

This example shows how to enable RXCRC port error detection on port 3/1:

```
Console> (enable) set port errordetection 3/1 rxcrc enable
Port(s)  3/1 set to errordetection rxcrc enable.
Console> (enable) show port errordetection 3/1
Port   Rxcrc    Txcrc
-----  -------- --------
 3/1   enabled  disabled
Console> (enable)
```

Configuring Redundant Flex Links

The redundant flex links provide an alternative solution to the Spanning Tree Protocol (STP) by allowing users to disable STP and still provide link-level redundancy. The redundant flex links provide a link backup capability with rapid switchover redundancy. Flex link redundancy allows you to specify two ports to form the redundant link capability. You configure one port as the active port and the other port as the backup or *peer* port. The active port is in the forwarding state while the backup port is in the blocking state and does not allow traffic to pass through.

When the active port of the flex links experiences a failure, the MAC addresses are flushed and flooded. The backup port of the flex links learns the MAC addresses and restores connectivity. The failover convergence time depends on the number of VLANs and the number of MAC addresses. You cannot enable STP on the flex-link ports but you can run STP on other ports in the switch.

Tip We recommend that you use redundant flex links for configurations that have multiple Layer 2 access switches with common VLANs that are connected to a Layer 2 concentrator switch through two uplink ports.

These sections describe how to configure redundant flex links on the Catalyst 6500 series switches:

- Redundant Flex Link Configuration Guidelines and Restrictions, page 4-17
- Specifying the Active and Backup Ports for the Flex Links, page 4-18
- Displaying the Port Configuration of the Flex Links, page 4-18
- Clearing the Port Configuration of the Flex Links, page 4-18

Redundant Flex Link Configuration Guidelines and Restrictions

This section describes the guidelines and restrictions for configuring redundant flex links:

- The maximum number of flex-link pairs (one active port and one backup port) is 16 per switch.
- Flex-link ports cannot be part of an EtherChannel.
- STP—Flex-link ports do not join STP operations. Flex-link ports do not generate STP bridge protocol data units (BPDUs) and they drop all received BPDUs.
- Switched Port Analyzer (SPAN)—SPAN works with flex-link ports.
- VLAN Trunk Protocol (VTP)—As VTP pruning requires working with STP, it does not work on flex-link ports.
- Internet Group Management Protocol (IGMP)—IGMP works with flex-link ports.
- Dynamic Trunking Protocol (DTP)—DTP can run on a flex-link port.

- Redundant flex links are for simple access topologies (two uplinks from a leaf node). You need to make sure that there is a loop-free path from the wiring closet to the access network. Unlike STP, the flex-link port is not designed to detect loops.
- Deploying STP in the core network while running flex-link redundancy on the edge is an acceptable configuration.
- The flex links converge faster only if the directly connected link fails. Any other failure in the network sees no improvement from the flex-link fast convergence.

Specifying the Active and Backup Ports for the Flex Links

To specify an active port and a backup port (peer) for the flex links, perform this task in privileged mode:

Task	Command
Specify an active port and a backup port (peer) for the flex links.	**set port flexlink** *mod/port* **peer** *mod/port*

This example shows how to specify port 3/48 as the flex-link active port and port 3/47 as the flex-link backup (peer) port:

```
Console> (enable) set port flexlink 3/48 peer 3/47
Flexlink is successfully set on the port 3/48 and 3/47
Console> (enable)
```

Displaying the Port Configuration of the Flex Links

To display information about the flex-link port configuration, perform this task in normal mode:

Task	Command
Display information about the flex-link port configuration.	**show port flexlink** [*mod* | *mod/port*]

This example shows how to display information about all the flex-link ports that are configured on the switch:

```
Console> (enable) show port flexlink
Port   State       Peer port  State
-----  ----------  ---------  ----------
 3/47  linkdown    3/48       active
 3/48  active      3/47       linkdown
Console> (enable)
```

Clearing the Port Configuration of the Flex Links

To clear the port configuration of the flex links, perform this task in privileged mode:

Task	Command
Clear the port configuration of the flex links.	**clear port flexlink** *mod/port* [**peer** *mod/port*]

This example shows how to clear port 3/48 as the flex-link active port and port 3/47 as the flex-link backup (peer) port:

```
Console> (enable) clear port flexlink 3/48 peer 3/47
Port 3/48 and 3/47 flexlink pair cleared
Console> (enable)
```

Configuring Jumbo Frames

These sections describe how to configure jumbo frames:

- Configuring Jumbo Frames on the Supervisor Engine, page 4-19
- Configuring Jumbo Frames on the MSFC2, page 4-20

Configuring Jumbo Frames on the Supervisor Engine

When you enable jumbo frames on a port, the port can switch large (or *jumbo*) frames. This feature is useful in optimizing server-to-server performance. The default maximum transmission unit (MTU) frame size is 1548 bytes for all Ethernet ports. By enabling jumbo frames on a port, the MTU size is increased to 9216 bytes.

To enable jumbo frames on a per-port basis, follow these guidelines:

Note The WS-X6148 and WS-X6548 GE-TX modules do not support jumbo frames.

- Jumbo frames are supported on the following:
  - All Ethernet ports
  - Trunk ports
  - EtherChannels
  - sc0 interface (jumbo frames are passed through the sc0 interface as a nonconfigurable default; no CLI configuration is necessary)
- These switching modules support a maximum ingress frame size of 8092 bytes:
  - WS-6516-GE-TX when operating at 100 Mbps. At 10 Mbps and 1000 Mbps, the module supports the jumbo frame default of 9216 bytes.
  - WS-X6148-RJ-45, WS-X6148-RJ-45V, WS-X6148-RJ21, and WS-X6148-RJ21V.
  - WS-X6248-RJ-45, WS-X6248A-RJ-45, WS-X6248-TEL, and WS-X6248A-TEL.
  - WS-X6348-RJ-45, WS-X6348-RJ45V, WS-X6348-RJ-21, and WX-X6348-RJ21V.

  When jumbo frame support is configured, these modules drop ingress frames that are larger than 8092 bytes.
- The WS-X6548-RJ-21 and WS-X6548-RJ-45 modules use different hardware at the PHY level and support the full jumbo frame default value of 9216 bytes.
- Jumbo frames are supported on all Optical Services Modules (OSMs).
- Jumbo frames are not supported on ATM modules (WS-X6101-OC12-SMF/MMF).

- The Multilayer Switching Feature Card 2 (MSFC2) supports jumbo frame routing with Cisco IOS Release 12.1(2)E and later releases.
- The Multilayer Switching Feature Card (MSFC) and the Multilayer Switch Module (MSM) do not support jumbo frame routing; if jumbo frames are sent to these routers, router performance is significantly degraded.

Note Occasionally, you might see a "Jumbo frames inconsistent state" message for a port or multiple ports after entering the **show port jumbo** command. If you see this message, enter the **set port jumbo** command to reenable the ports.

To enable jumbo frames on an Ethernet port, perform this task in privileged mode:

	Task	Command
Step 1	Enable jumbo frames.	**set port jumbo** *mod/port* **enable**
Step 2	Verify the port configuration.	**show port jumbo**

This example shows how to enable jumbo frames on a port and verify the configuration:

```
Console> (enable) set port jumbo 2/1 enable
Jumbo frames enabled on port 2/1
Console> (enable) show port jumbo
Jumbo frames MTU size is 9216 bytes
Jumbo frames enabled on port(s) 2/1
```

To disable jumbo frames on an Ethernet port, perform this task in privileged mode:

	Task	Command
Step 1	Disable jumbo frames.	**set port jumbo** *mod/port* **disable**
Step 2	Verify the port configuration.	**show port jumbo**

This example shows how to disable jumbo frames on a port:

```
Console> (enable) set port jumbo 2/1 disable
Jumbo frames disabled on port 2/1
Console> (enable)
```

Configuring Jumbo Frames on the MSFC2

With an MSFC2, you can configure the MTU size on VLAN interfaces to support jumbo frame routing.

Jumbo frames support only a single larger-than-default MTU size on the switch. Configuring a VLAN interface with an MTU size that is greater than the default automatically configures all other VLAN interfaces that have an MTU size that is greater than the default to the newly configured size. The VLAN interfaces that have not been changed from the default are not affected.

To configure the MTU value, perform this task:

	Task	Command
Step 1	Access VLAN interface configuration mode.	Router(config)# **interface vlan** *vlan_ID*
Step 2	Set the MTU size. The valid values are from 64 to 17952 bytes[1].	Router(config-if)# **mtu** *mtu_size*
Step 3	Verify the configuration.	Router# **show interface vlan 111**

1. Set the MTU size no larger than 9216, which is the size that is supported by the supervisor engine.

This example shows how to set the MTU size on a VLAN interface and verify the configuration:

```
Router(config)# interface vlan 111
Router(config-if)# mtu 9216
Router(config-if)# end
Router# show interface vlan 111
.
.
.
MTU 9216 bytes, BW 1000000 Kbit, DLY 10 usec,
.
.
.
Router#
```

Checking Connectivity

Use the **ping** and **traceroute** commands to test connectivity.

To check connectivity out a port, perform this task in privileged mode:

	Task	Command
Step 1	Ping a remote host that is located out the port that you want to test.	**ping** [**-s**] *host* [*packet_size*] [*packet_count*]
Step 2	Trace the hop-by-hop route of packets from the switch to a remote host that is located out the port that you want to test.	**traceroute** *host*
Step 3	If the host is unresponsive, check the IP address and default gateway that are configured on the switch.	**show interface** **show ip route**

This example shows how to ping a remote host and trace the hop-by-hop path of packets through the network using **traceroute**:

```
Console> (enable) ping somehost
somehost is alive
Console> (enable) traceroute somehost
traceroute to somehost.company.com (10.1.2.3), 30 hops max, 40 byte packets
 1 engineering-1.company.com (173.31.192.206) 2 ms 1 ms 1 ms
 2 engineering-2.company.com (173.31.196.204) 2 ms 3 ms 2 ms
 3 gateway_a.company.com (173.16.1.201) 6 ms 3 ms 3 ms
 4 somehost.company.com (10.1.2.3) 3 ms * 2 ms
Console> (enable)
```