



## Configuring Optional STP Features

---

This chapter describes how to configure optional STP features.



### Note

- For complete syntax and usage information for the commands used in this chapter, refer to the *Cisco IOS Master Command List*, Release 12.2SX at this URL:  
[http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all\\_book.html](http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html)
  - For information on configuring the spanning tree protocol (STP), see [Chapter 20, “Configuring STP and Prestandard IEEE 802.1s MST.”](#)
- 

This chapter consists of these sections:

- [Understanding How PortFast Works](#), page 21-2
- [Understanding How BPDU Guard Works](#), page 21-2
- [Understanding How PortFast BPDU Filtering Works](#), page 21-2
- [Understanding How UplinkFast Works](#), page 21-3
- [Understanding How BackboneFast Works](#), page 21-4
- [Understanding How EtherChannel Guard Works](#), page 21-6
- [Understanding How Root Guard Works](#), page 21-7
- [Understanding How Loop Guard Works](#), page 21-7
- [Enabling PortFast](#), page 21-8
- [Enabling PortFast BPDU Filtering](#), page 21-10
- [Enabling BPDU Guard](#), page 21-12
- [Enabling UplinkFast](#), page 21-12
- [Enabling BackboneFast](#), page 21-13
- [Enabling EtherChannel Guard](#), page 21-14
- [Enabling Root Guard](#), page 21-14
- [Enabling Loop Guard](#), page 21-15

**Tip**

For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

[http://www.cisco.com/en/US/products/hw/switches/ps708/tsd\\_products\\_support\\_series\\_home.html](http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html)

[Participate in the Technical Documentation Ideas forum](#)

## Understanding How PortFast Works

STP PortFast causes a Layer 2 LAN port configured as an access port to enter the forwarding state immediately, bypassing the listening and learning states. You can use PortFast on Layer 2 access ports connected to a single workstation or server to allow those devices to connect to the network immediately, instead of waiting for STP to converge. Interfaces connected to a single workstation or server should not receive bridge protocol data units (BPDUs). When configured for PortFast, a port is still running the spanning tree protocol. A PortFast enabled port can immediately transition to the blocking state if necessary (this could happen on receipt of a superior BPDU). PortFast can be enabled on trunk ports. PortFast can have an operational value that is different from the configured value.

**Caution**

Because the purpose of PortFast is to minimize the time that access ports must wait for STP to converge, it should only be used on access ports. If you enable PortFast on a port connected to a switch, you might create a temporary bridging loop.

## Understanding How BPDU Guard Works

When enabled on a port, BPDU Guard shuts down a port that receives a BPDU. When configured globally, BPDU Guard is only effective on ports in the operational PortFast state. In a valid configuration, PortFast Layer 2 LAN interfaces do not receive BPDUs. Reception of a BPDU by a PortFast Layer 2 LAN interface signals an invalid configuration, such as connection of an unauthorized device. BPDU Guard provides a secure response to invalid configurations, because the administrator must manually put the Layer 2 LAN interface back in service. BPDU Guard can be configured at the interface level. When configured at the interface level, BPDU Guard shuts the port down as soon as the port receives a BPDU, regardless of the PortFast configuration.

**Note**

When enabled globally, BPDU Guard applies to all interfaces that are in an operational PortFast state.

## Understanding How PortFast BPDU Filtering Works

PortFast BPDU filtering allows the administrator to prevent the system from sending or even receiving BPDUs on specified ports.

When configured globally, PortFast BPDU filtering applies to all operational PortFast ports. Ports in an operational PortFast state are supposed to be connected to hosts, that typically drop BPDUs. If an operational PortFast port receives a BPDU, it immediately loses its operational PortFast status. In that case, PortFast BPDU filtering is disabled on this port and STP resumes sending BPDUs on this port.

PortFast BPDU filtering can also be configured on a per-port basis. When PortFast BPDU filtering is explicitly configured on a port, it does not send any BPDUs and drops all BPDUs it receives.

**Caution**

Configuring PortFast BPDU filtering on a port that is not connected to a host can result in bridging loops.

When you enable PortFast BPDU filtering globally and set the port configuration as the default for PortFast BPDU filtering (see the “[Enabling PortFast BPDU Filtering](#)” section on page 21-10), then PortFast enables or disables PortFast BPDU filtering.

If the port configuration is not set to default, then the PortFast configuration will not affect PortFast BPDU filtering. [Table 21-1](#) lists all the possible PortFast BPDU filtering combinations. PortFast BPDU filtering allows access ports to move directly to the forwarding state as soon as the end hosts are connected.

**Table 21-1 PortFast BPDU Filtering Port Configurations**

Per-Port Configuration	Global Configuration	PortFast State	PortFast BPDU Filtering State
Default	Enable	Enable	Enable <sup>1</sup>
Default	Enable	Disable	Disable
Default	Disable	Not applicable	Disable
Disable	Not applicable	Not applicable	Disable
Enable	Not applicable	Not applicable	Enable

1. The port transmits at least 10 BPDUs. If this port receives any BPDUs, then PortFast and PortFast BPDU filtering are disabled.

## Understanding How UplinkFast Works

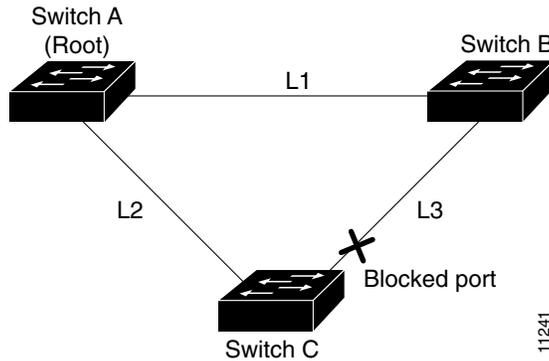
UplinkFast provides fast convergence after a direct link failure and achieves load balancing between redundant Layer 2 links using uplink groups. An uplink group is a set of Layer 2 LAN interfaces (per VLAN), only one of which is forwarding at any given time. Specifically, an uplink group consists of the root port (which is forwarding) and a set of blocked ports, except for self-looping ports. The uplink group provides an alternate path in case the currently forwarding link fails.

**Note**

UplinkFast is most useful in wiring-closet switches. This feature may not be useful for other types of applications.

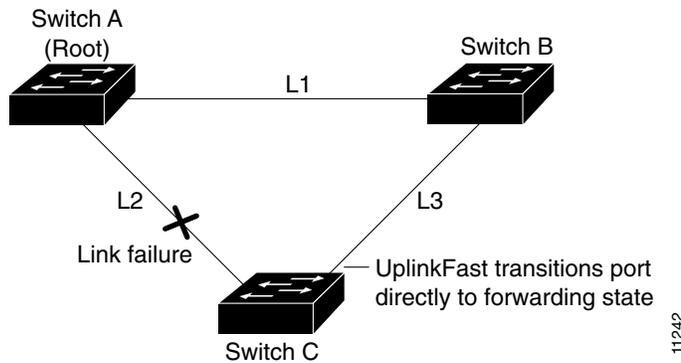
[Figure 21-1](#) shows an example topology with no link failures. Switch A, the root bridge, is connected directly to Switch B over link L1 and to Switch C over link L2. The Layer 2 LAN interface on Switch C that is connected directly to Switch B is in the blocking state.

Figure 21-1 UplinkFast Example Before Direct Link Failure



If Switch C detects a link failure on the currently active link L2 on the root port (a *direct* link failure), UplinkFast unblocks the blocked port on Switch C and transitions it to the forwarding state without going through the listening and learning states, as shown in Figure 21-2. This switchover takes approximately one to five seconds.

Figure 21-2 UplinkFast Example After Direct Link Failure



## Understanding How BackboneFast Works

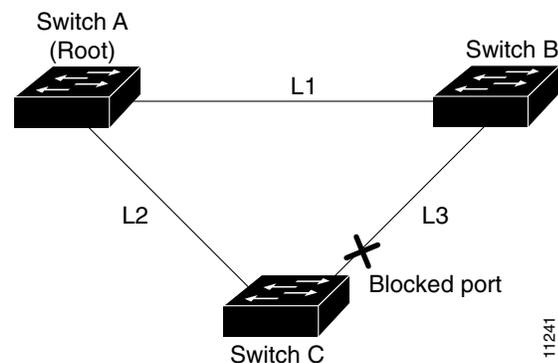
BackboneFast is initiated when a root port or blocked port on a network device receives inferior BPDUs from its designated bridge. An inferior BPDU identifies one network device as both the root bridge and the designated bridge. When a network device receives an inferior BPDU, it indicates that a link to which the network device is not directly connected (an *indirect* link) has failed (that is, the designated bridge has lost its connection to the root bridge). Under normal STP rules, the network device ignores inferior BPDUs for the configured maximum aging time, as specified by the STP **max-age** command.

The network device tries to determine if it has an alternate path to the root bridge. If the inferior BPDU arrives on a blocked port, the root port and other blocked ports on the network device become alternate paths to the root bridge. (Self-looped ports are not considered alternate paths to the root bridge.) If the inferior BPDU arrives on the root port, all blocked ports become alternate paths to the root bridge. If the inferior BPDU arrives on the root port and there are no blocked ports, the network device assumes that it has lost connectivity to the root bridge, causes the maximum aging time on the root to expire, and becomes the root bridge according to normal STP rules.

If the network device has alternate paths to the root bridge, it uses these alternate paths to transmit a new kind of Protocol Data Unit (PDU) called the Root Link Query PDU. The network device sends the Root Link Query PDU out all alternate paths to the root bridge. If the network device determines that it still has an alternate path to the root, it causes the maximum aging time to expire on the ports on which it received the inferior BPDU. If all the alternate paths to the root bridge indicate that the network device has lost connectivity to the root bridge, the network device causes the maximum aging times on the ports on which it received an inferior BPDU to expire. If one or more alternate paths can still connect to the root bridge, the network device makes all ports on which it received an inferior BPDU its designated ports and moves them out of the blocking state (if they were in the blocking state), through the listening and learning states, and into the forwarding state.

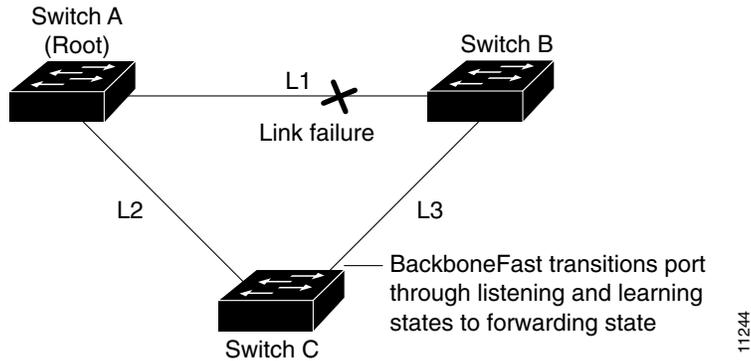
Figure 21-3 shows an example topology with no link failures. Switch A, the root bridge, connects directly to Switch B over link L1 and to Switch C over link L2. The Layer 2 LAN interface on Switch C that connects directly to Switch B is in the blocking state.

**Figure 21-3 BackboneFast Example Before Indirect Link Failure**



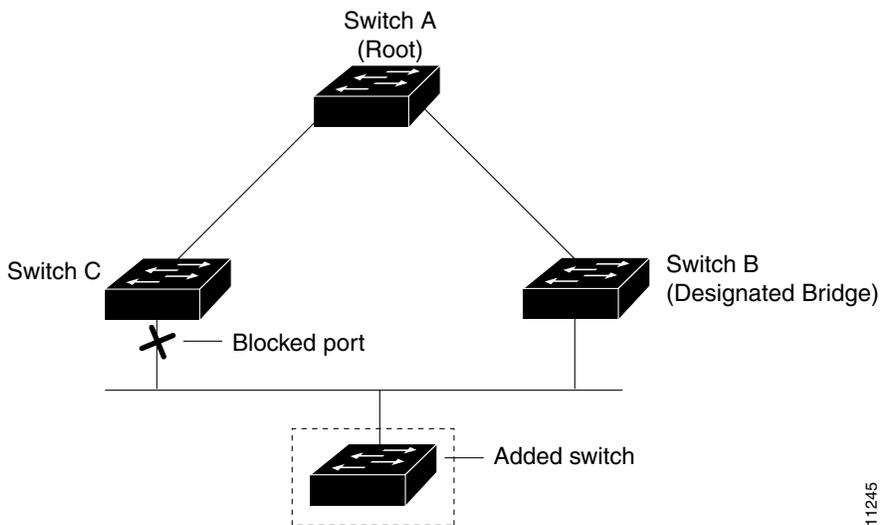
If link L1 fails, Switch C cannot detect this failure because it is not connected directly to link L1. However, because Switch B is directly connected to the root bridge over L1, it detects the failure and elects itself the root and begins sending BPDUs to Switch C indicating itself as the root. When Switch C receives the inferior BPDUs from Switch B, Switch C infers that an indirect failure has occurred. At that point, BackboneFast allows the blocked port on Switch C to move immediately to the listening state without waiting for the maximum aging time for the port to expire. BackboneFast then transitions the Layer 2 LAN interface on Switch C to the forwarding state, providing a path from Switch B to Switch A. This switchover takes approximately 30 seconds, twice the Forward Delay time if the default Forward Delay time of 15 seconds is set. Figure 21-4 shows how BackboneFast reconfigures the topology to account for the failure of link L1.

Figure 21-4 BackboneFast Example After Indirect Link Failure



If a new network device is introduced into a shared-medium topology as shown in Figure 21-5, BackboneFast is not activated because the inferior BPDUs did not come from the recognized designated bridge (Switch B). The new network device begins sending inferior BPDUs that indicate that it is the root bridge. However, the other network devices ignore these inferior BPDUs and the new network device learns that Switch B is the designated bridge to Switch A, the root bridge.

Figure 21-5 Adding a Network Device in a Shared-Medium Topology



## Understanding How EtherChannel Guard Works

EtherChannel guard detects a misconfigured EtherChannel where interfaces on the Catalyst 6500 series switch are configured as an EtherChannel while interfaces on the other device are not, or not all the interfaces on the other device are in the same EtherChannel.

In response to misconfiguration detected on the other device, EtherChannel guard puts interfaces on the Catalyst 6500 series switch into the errdisabled state.

## Understanding How Root Guard Works

The STP root guard feature prevents a port from becoming root port or blocked port. If a port configured for root guard receives a superior BPDUs, the port immediately goes to the root-inconsistent (blocked) state.

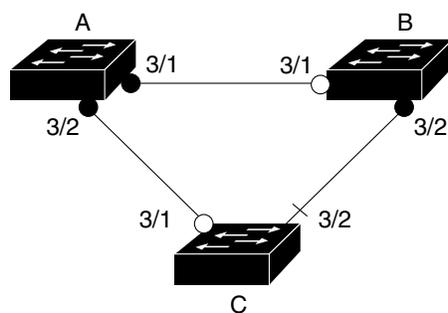
## Understanding How Loop Guard Works

Loop guard helps prevent bridging loops that could occur because of a uni-directional link failure on a point-to-point link. When enabled globally, the loop guard applies to all point-to-point ports on the system. Loop guard detects root ports and blocked ports and ensures that they keep receiving BPDUs from their designated port on the segment. If a loop guard enabled root or blocked port stop a receiving BPDUs from its designated port, it transitions to the loop-inconsistent blocking state, assuming there is a physical link error on this port. The port recovers from this loop-inconsistent state as soon as it receives a BPDU.

You can enable loop guard on a per-port basis. When you enable loop guard, it is automatically applied to all of the active instances or VLANs to which that port belongs. When you disable loop guard, it is disabled for the specified ports. Disabling loop guard moves all loop-inconsistent ports to the listening state.

If you enable loop guard on a channel and the first link becomes unidirectional, loop guard blocks the entire channel until the affected port is removed from the channel. [Figure 21-6](#) shows loop guard in a triangle switch configuration.

**Figure 21-6 Triangle Switch Configuration with Loop Guard**



- Designated port
- Root port
- + Alternate port

55772

[Figure 21-6](#) illustrates the following configuration:

- Switches A and B are distribution switches.
- Switch C is an access switch.
- Loop guard is enabled on ports 3/1 and 3/2 on Switches A, B, and C.

Enabling loop guard on a root switch has no effect but provides protection when a root switch becomes a nonroot switch.

When using loop guard, follow these guidelines:

- You cannot enable loop guard on PortFast-enabled ports.
- You cannot enable loop guard if root guard is enabled.

Loop guard interacts with other features as follows:

- Loop guard does not affect the functionality of UplinkFast or BackboneFast.
- Enabling loop guard on ports that are not connected to a point-to-point link will not work.
- Root guard forces a port to be always designated as the root port. Loop guard is effective only if the port is a root port or an alternate port. You cannot enable loop guard and root guard on a port at the same time.
- Loop guard uses the ports known to spanning tree. Loop guard can take advantage of logical ports provided by the Port Aggregation Protocol (PAgP). However, to form a channel, all the physical ports grouped in the channel must have compatible configurations. PAgP enforces uniform configurations of root guard or loop guard on all the physical ports to form a channel.

These caveats apply to loop guard:

- Spanning tree always chooses the first operational port in the channel to send the BPDUs. If that link becomes unidirectional, loop guard blocks the channel, even if other links in the channel are functioning properly.
- If a set of ports that are already blocked by loop guard are grouped together to form a channel, spanning tree loses all the state information for those ports and the new channel port may obtain the forwarding state with a designated role.
- If a channel is blocked by loop guard and the channel breaks, spanning tree loses all the state information. The individual physical ports may obtain the forwarding state with the designated role, even if one or more of the links that formed the channel are unidirectional.



**Note** You can enable UniDirectional Link Detection (UDLD) to help isolate the link failure. A loop may occur until UDLD detects the failure, but loop guard will not be able to detect it.

- Loop guard has no effect on a disabled spanning tree instance or a VLAN.

## Enabling PortFast



### Caution

Use PortFast *only* when connecting a single end station to a Layer 2 access port. Otherwise, you might create a network loop.

To enable PortFast on a Layer 2 access port, perform this task:

	Command	Purpose
<b>Step 1</b>	Router(config)# <b>interface</b> {type <sup>1</sup> slot/port}   {port-channel port_channel_number}	Selects a port to configure.
<b>Step 2</b>	Router(config-if)# <b>spanning-tree portfast</b>	Enables PortFast on a Layer 2 access port connected to a single workstation or server.
<b>Step 3</b>	Router(config-if)# <b>spanning-tree portfast default</b>	Enables PortFast.

	Command	Purpose
Step 4	Router(config-if)# <b>end</b>	Exits configuration mode.
Step 5	Router# <b>show running interface</b> {type <sup>1</sup> slot/port}   {port-channel port_channel_number}	Verifies the configuration.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to enable PortFast on Fast Ethernet interface 5/8:

```
Router# configure terminal
Router(config)# interface fastethernet 5/8
Router(config-if)# spanning-tree portfast
Router(config-if)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show running-config interface fastethernet 5/8
Building configuration...
```

```
Current configuration:
!
interface FastEthernet5/8
  no ip address
  switchport
  switchport access vlan 200
  switchport mode access
  spanning-tree portfast
end
```

```
Router#
```

To enable the default PortFast configuration, perform this task:

	Command	Purpose
Step 1	Router(config)# <b>spanning-tree portfast default</b>	Configures the PortFast default.
Step 2	Router(config)# <b>show spanning-tree summary totals</b>	Verifies the global configuration.
Step 3	Router(config)# <b>show spanning-tree interface x detail</b>	Verifies the effect on a specific port.
Step 4	Router(config-if)# <b>spanning-tree portfast trunk</b>	Enables the PortFast trunk on a port
Step 5	Router# <b>show spanning-tree interface fastEthernet x detail</b>	Verifies the configuration.

This example shows how to enable the default PortFast configuration:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# spanning-tree portfast default
Router(config)# ^Z
```

```
Root bridge for:VLAN0010
EtherChannel misconfiguration guard is enabled
Extended system ID is disabled
Portfast is enabled by default
PortFast BPDU Guard is disabled by default
Portfast BPDU Filter is disabled by default
Loopguard is disabled by default
UplinkFast is disabled
BackboneFast is disabled
Pathcost method used is long
```

```

Name                    Blocking Listening Learning Forwarding STP Active
-----
VLAN0001                0          0          0          1          1
VLAN0010                0          0          0          2          2
-----
2 vlans                 0          0          0          3          3
Router#

```

```

Router# show spanning-tree interface fastEthernet 4/4 detail
Port 196 (FastEthernet4/4) of VLAN0010 is forwarding
  Port path cost 1000, Port priority 160, Port Identifier 160.196.
  Designated root has priority 32768, address 00d0.00b8.140a
  Designated bridge has priority 32768, address 00d0.00b8.140a
  Designated port id is 160.196, designated path cost 0
  Timers:message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state:1
  The port is in the portfast mode by default
  Link type is point-to-point by default
  BPDU:sent 10, received 0

```

```

Router(config-if)# spanning-tree portfast trunk
%Warning:portfast should only be enabled on ports connected to a single
  host. Connecting hubs, concentrators, switches, bridges, etc... to this
  interface when portfast is enabled, can cause temporary bridging loops.
  Use with CAUTION

```

```

Router(config-if)# ^Z

```

```

Router# show spanning-tree interface fastEthernet 4/4 detail
Port 196 (FastEthernet4/4) of VLAN0010 is forwarding
  Port path cost 1000, Port priority 160, Port Identifier 160.196.
  Designated root has priority 32768, address 00d0.00b8.140a
  Designated bridge has priority 32768, address 00d0.00b8.140a
  Designated port id is 160.196, designated path cost 0
  Timers:message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state:1
  The port is in the portfast mode by portfast trunk configuration
  Link type is point-to-point by default
  BPDU:sent 30, received 0
Router#

```

## Enabling PortFast BPDU Filtering

These sections describe how to configure PortFast BPDU filtering.

To enable PortFast BPDU filtering globally, perform this task:

	Command	Purpose
Step 1	Router(config)# <b>spanning-tree portfast bpdufilter default</b>	Enables BPDU filtering globally on the switch.
Step 2	Router# <b>show spanning-tree summary totals</b>	Verifies the configuration.

BPDU filtering is set to default on each port. This example shows how to enable PortFast BPDU filtering on the port and verify the configuration in PVST+ mode:

**Note**

For PVST+ information, see [Chapter 20, “Configuring STP and Prestandard IEEE 802.1s MST.”](#)

```
Router(config)# spanning-tree portfast bpdupfilter default
Router(config)# ^Z
```

```
Router# show spanning-tree summary totals
Root bridge for:VLAN0010
EtherChannel misconfiguration guard is enabled
Extended system ID   is disabled
Portfast              is enabled by default
PortFast BPDU Guard  is disabled by default
Portfast BPDU Filter is enabled by default
Loopguard             is disabled by default
UplinkFast           is disabled
BackboneFast         is disabled
Pathcost method used is long
```

```
Name                               Blocking Listening Learning Forwarding STP Active
-----
2 vlans                            0           0           0           3           3
Router#
```

To enable PortFast BPDU filtering on a nontrunking port, perform this task:

	Command	Purpose
<b>Step 1</b>	Router(config)# <b>interface fastEthernet 4/4</b>	Selects the interface to configure.
<b>Step 2</b>	Router(config-if)# <b>spanning-tree bpdupfilter enable</b>	Enables BPDU filtering.
<b>Step 3</b>	Router# <b>show spanning-tree interface fastEthernet 4/4</b>	Verifies the configuration.

This example shows how to enable PortFast BPDU filtering on a nontrunking port:

```
Router(config)# interface fastEthernet 4/4
Router(config-if)# spanning-tree bpdupfilter enable
Router(config-if)# ^Z
```

```
Router# show spanning-tree interface fastEthernet 4/4
```

```
Vlan          Role Sts Cost      Prio.Nbr Status
-----
VLAN0010      Desg FWD 1000      160.196 Edge P2p
Router# show spanning-tree interface fastEthernet 4/4 detail
Router# show spanning-tree interface fastEthernet 4/4 detail
Port 196 (FastEthernet4/4) of VLAN0010 is forwarding
  Port path cost 1000, Port priority 160, Port Identifier 160.196.
  Designated root has priority 32768, address 00d0.00b8.140a
  Designated bridge has priority 32768, address 00d0.00b8.140a
  Designated port id is 160.196, designated path cost 0
  Timers:message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state:1
  The port is in the portfast mode by portfast trunk configuration
  Link type is point-to-point by default
  Bpdu filter is enabled
  BPDU:sent 0, received 0
Router#
```

## Enabling BPDU Guard

To enable BPDU Guard globally, perform this task:

	Command	Purpose
Step 1	Router(config)# <b>spanning-tree portfast bpduguard default</b>	Enables BPDU Guard globally.
	Router(config)# <b>no spanning-tree portfast bpduguard default</b>	Disables BPDU Guard globally.
Step 2	Router(config)# <b>end</b>	Exits configuration mode.
Step 3	Router# <b>show spanning-tree summary totals</b>	Verifies the configuration.

This example shows how to enable BPDU Guard:

```
Router# configure terminal
Router(config)# spanning-tree portfast bpduguard
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree summary totals default
Root bridge for:VLAN0010
EtherChannel misconfiguration guard is enabled
Extended system ID is disabled
Portfast is enabled by default
PortFast BPDU Guard is disabled by default
Portfast BPDU Filter is enabled by default
Loopguard is disabled by default
UplinkFast is disabled
BackboneFast is disabled
Pathcost method used is long
```

```
Name Blocking Listening Learning Forwarding STP Active
-----
2 vlans 0 0 0 3 3
Router#
```

## Enabling UplinkFast

UplinkFast increases the bridge priority to 49152 and adds 3000 to the STP port cost of all Layer 2 LAN interfaces on the Catalyst 6500 series switch, decreasing the probability that the switch will become the root bridge. UplinkFast cannot be enabled on VLANs that have been configured for bridge priority. To enable UplinkFast on a VLAN with bridge priority configured, restore the bridge priority on the VLAN to the default value by entering a **no spanning-tree vlan *vlan\_ID* priority** command in global configuration mode.



### Note

When you enable UplinkFast, it affects all VLANs on the Catalyst 6500 series switch. You cannot configure UplinkFast on an individual VLAN.

To enable UplinkFast, perform this task:

	Command	Purpose
<b>Step 1</b>	Router(config)# <b>spanning-tree uplinkfast</b>	Enables UplinkFast.
	Router(config)# <b>spanning-tree uplinkfast</b> [ <b>max-update-rate</b> <i>max_update_rate</i> ]	Enables UplinkFast with an update rate in seconds.
	Router(config)# <b>no spanning-tree uplinkfast</b> <b>max-update-rate</b>	Reverts to the default rate.
	Router(config)# <b>no spanning-tree uplinkfast</b>	Disables UplinkFast.
<b>Step 2</b>	Router(config)# <b>end</b>	Exits configuration mode.
<b>Step 3</b>	Router# <b>show spanning-tree vlan</b> <i>vlan_ID</i>	Verifies that UplinkFast is enabled.

This example shows how to enable UplinkFast:

```
Router# configure terminal
Router(config)# spanning-tree uplinkfast
Router(config)# exit
Router#
```

This example shows how to enable UplinkFast with an update rate of 400 packets per second:

```
Router# configure terminal
Router(config)# spanning-tree uplinkfast
Router(config)# spanning-tree uplinkfast max-update-rate 400
Router(config)# exit
Router#
```

This example shows how to verify that UplinkFast is enabled:

```
Router# show spanning-tree uplinkfast
UplinkFast is enabled
Router#
```

## Enabling BackboneFast



### Note

BackboneFast operates correctly only when enabled on all network devices in the network. BackboneFast is not supported on Token Ring VLANs. This feature is supported for use with third-party network devices.

To enable BackboneFast, perform this task:

	Command	Purpose
<b>Step 1</b>	Router(config)# <b>spanning-tree backbonefast</b>	Enables BackboneFast.
	Router(config)# <b>no spanning-tree backbonefast</b>	Disables BackboneFast.
<b>Step 2</b>	Router(config)# <b>end</b>	Exits configuration mode.
<b>Step 3</b>	Router# <b>show spanning-tree vlan</b> <i>vlan_ID</i>	Verifies that BackboneFast is enabled.

This example shows how to enable BackboneFast:

```
Router# configure terminal
```

```
Router(config)# spanning-tree backbonefast
Router(config)# end
Router#
```

This example shows how to verify that BackboneFast is enabled:

```
Router# show spanning-tree backbonefast
BackboneFast is enabled

BackboneFast statistics
-----
Number of transition via backboneFast (all VLANs) : 0
Number of inferior BPDUs received (all VLANs)    : 0
Number of RLQ request PDUs received (all VLANs)  : 0
Number of RLQ response PDUs received (all VLANs) : 0
Number of RLQ request PDUs sent (all VLANs)     : 0
Number of RLQ response PDUs sent (all VLANs)    : 0
Router#
```

## Enabling EtherChannel Guard

To enable EtherChannel guard, perform this task:

	Command	Purpose
Step 1	Router(config)# <b>spanning-tree etherchannel guard misconfig</b>	Enables EtherChannel guard.
	Router(config)# <b>no spanning-tree etherchannel guard misconfig</b>	Disables EtherChannel guard.
Step 2	Router(config)# <b>end</b>	Exits configuration mode.
Step 3	Router# <b>show spanning-tree summary   include EtherChannel</b>	Verifies that EtherChannel guard is enabled.

This example shows how to enable EtherChannel guard:

```
Router# configure terminal
Router(config)# spanning-tree etherchannel guard misconfig
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show spanning-tree summary | include EtherChannel
EtherChannel misconfiguration guard is enabled
```

To display the interfaces that are in the errdisable state, enter the **show interface status err-disable** command.

After the misconfiguration has been cleared, interfaces in the errdisable state might automatically recover. To manually return a port to service, enter a **shutdown** and then a **no shutdown** command for the interface.

## Enabling Root Guard

To enable root guard, perform this task:

	Command	Purpose
Step 1	Router(config)# <b>interface</b> {type <sup>1</sup> slot/port}   {port-channel port_channel_number}	Selects a port to configure.
Step 2	Router(config-if)# <b>spanning-tree guard root</b>	Enables root guard.
	Router(config-if)# <b>no spanning-tree guard root</b>	Disables root guard.
Step 3	Router(config-if)# <b>end</b>	Exits configuration mode.
Step 4	Router# <b>show spanning-tree</b> Router# <b>show running interface</b> {type <sup>1</sup> slot/port}   {port-channel port_channel_number}	Verifies the configuration.

1. type = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

To display ports that are in the root-inconsistent state, enter the **show spanning-tree inconsistentports** command.

## Enabling Loop Guard

To enable loop guard globally on the switch, perform this task:

	Command	Purpose
Step 1	Router(config)# <b>spanning-tree loopguard default</b>	Enables loop guard globally on the switch.
Step 2	Router(config)# <b>end</b>	Exits configuration mode.
Step 3	Router# <b>show spanning-tree interface 4/4 detail</b>	Verifies the configuration impact on a port.

This example shows how to enable loop guard globally:

```
Router# configure terminal
```

Enter configuration commands, one per line. End with CNTL/Z.

```
Router(config)# spanning-tree loopguard default
```

```
Router(config)# ^Z
```

```
Router# show spanning-tree interface fastEthernet 4/4 detail
```

```
Port 196 (FastEthernet4/4) of VLAN0010 is forwarding
  Port path cost 1000, Port priority 160, Port Identifier 160.196.
  Designated root has priority 32768, address 00d0.00b8.140a
  Designated bridge has priority 32768, address 00d0.00b8.140a
  Designated port id is 160.196, designated path cost 0
  Timers:message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state:1
  The port is in the portfast mode by portfast trunk configuration
  Link type is point-to-point by default
  Bpdu filter is enabled
  Loop guard is enabled by default on the port
  BPDU:sent 0, received 0
```

To enable loop guard on a port, perform this task:

	Command	Purpose
Step 1	Router(config)# <b>interface</b> {type <sup>1</sup> slot/port}   {port-channel port_channel_number}	Selects a port to configure.
Step 2	Router(config-if)# <b>spanning-tree guard loop</b>	Configures loop guard.
Step 3	Router(config)# <b>end</b>	Exits configuration mode.
Step 4	Router# <b>show spanning-tree interface 4/4 detail</b>	Verifies the configuration impact on that port.

1. *type* = ethernet, fastethernet, gigabitethernet, or tengigabitethernet

This example shows how to enable loop guard:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface fastEthernet 4/4
Router(config-if)# spanning-tree guard loop
Router(config-if)# ^Z
```

This example shows how to verify the configuration:

```
Router# show spanning-tree interface fastEthernet 4/4 detail
Port 196 (FastEthernet4/4) of VLAN0010 is forwarding
  Port path cost 1000, Port priority 160, Port Identifier 160.196.
  Designated root has priority 32768, address 00d0.00b8.140a
  Designated bridge has priority 32768, address 00d0.00b8.140a
  Designated port id is 160.196, designated path cost 0
  Timers:message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state:1
  The port is in the portfast mode by portfast trunk configuration
  Link type is point-to-point by default
  Bpdu filter is enabled
  Loop guard is enabled on the port
  BPDU:sent 0, received 0
Router#
```



#### Tip

For additional information about Cisco Catalyst 6500 Series Switches (including configuration examples and troubleshooting information), see the documents listed on this page:

[http://www.cisco.com/en/US/products/hw/switches/ps708/tsd\\_products\\_support\\_series\\_home.html](http://www.cisco.com/en/US/products/hw/switches/ps708/tsd_products_support_series_home.html)

[Participate in the Technical Documentation Ideas forum](#)