Configuring Optional Spanning-Tree Features

Finding Feature Information

Your software release may not support all the features documented in this chapter. For the latest feature information and caveats, see the release notes for your platform and software release.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for the Optional Spanning-Tree Features

You can configure all of these features when your switch is running the per-VLAN spanning-tree plus (PVST+). You can configure only the noted features when your switch is running the Multiple Spanning Tree Protocol (MSTP) or the rapid per-VLAN spanning-tree plus (rapid-PVST+) protocol.

Restrictions for the Optional Spanning-Tree Features

You can configure the UplinkFast or the BackboneFast feature for rapid PVST+ or for the MSTP, but the feature remains disabled (inactive) until you change the spanning-tree mode to PVST+.

Information About Configuring the Optional Spanning-Tree Features

PortFast

PortFast immediately brings an interface configured as an access or trunk port to the forwarding state from a blocking state, bypassing the listening and learning states. You can use PortFast on interfaces connected to a single workstation or server, as shown in Figure 22-1, to allow those devices to immediately connect to the network, rather than waiting for the spanning tree to converge.

Interfaces connected to a single workstation or server should not receive bridge protocol data units (BPDUs). An interface with PortFast enabled goes through the normal cycle of spanning-tree status changes when the switch is restarted.

✎
Note Because the purpose of PortFast is to minimize the time interfaces must wait for spanning-tree to converge, it is effective only when used on interfaces connected to end stations. If you enable PortFast on an interface connecting to another switch, you risk creating a spanning-tree loop.

You can enable this feature by using the **spanning-tree portfast** interface configuration or the **spanning-tree portfast default** global configuration command.

Figure 22-1   PortFast-Enabled Interfaces
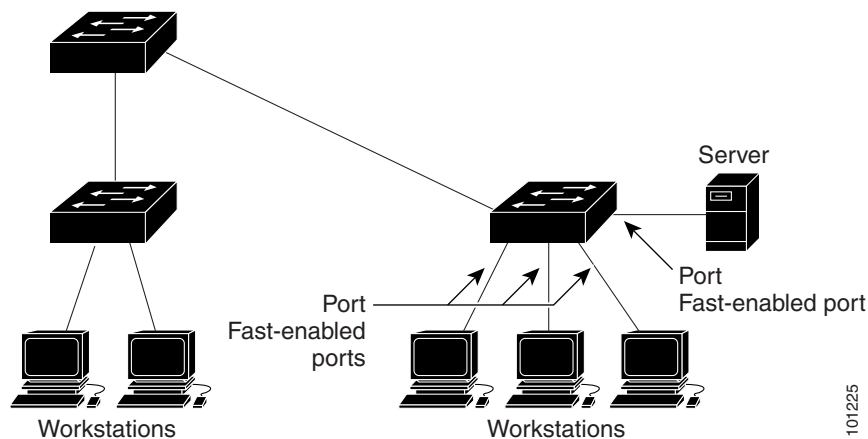
BPDU Guard

The BPDU guard feature can be globally enabled on the switch or can be enabled per port, but the feature operates with some differences.

At the global level, you enable BPDU guard on PortFast-enabled ports by using the **spanning-tree portfast bpduguard default** global configuration command. Spanning tree shuts down ports that are in a PortFast-operational state if any BPDU is received on them. In a valid configuration, PortFast-enabled ports do not receive BPDUs. Receiving a BPDU on a PortFast-enabled port means an invalid configuration, such as the connection of an unauthorized device, and the BPDU guard feature puts the port in the error-disabled state. When this happens, the switch shuts down the entire port on which the violation occurred.

To prevent the port from shutting down, you can use the **errdisable detect cause bpduguard shutdown vlan** global configuration command to shut down just the offending VLAN on the port where the violation occurred.

At the interface level, you enable BPDU guard on any port by using the **spanning-tree bpduguard enable** interface configuration command without also enabling the PortFast feature. When the port receives a BPDU, it is put in the error-disabled state.

The BPDU guard feature provides a secure response to invalid configurations because you must manually put the interface back in service. Use the BPDU guard feature in a service-provider network to prevent an access port from participating in the spanning tree.

BPDU Filtering

The BPDU filtering feature can be globally enabled on the switch or can be enabled per interface, but the feature operates with some differences.

At the global level, you can enable BPDU filtering on PortFast-enabled interfaces by using the **spanning-tree portfast bpdufilter default** global configuration command. This command prevents interfaces that are in a PortFast-operational state from sending or receiving BPDUs. The interfaces still send a few BPDUs at link-up before the switch begins to filter outbound BPDUs. You should globally enable BPDU filtering on a switch so that hosts connected to these interfaces do not receive BPDUs. If a BPDU is received on a PortFast-enabled interface, the interface loses its PortFast-operational status, and BPDU filtering is disabled.

At the interface level, you can enable BPDU filtering on any interface by using the **spanning-tree bpdufilter enable** interface configuration command without also enabling the PortFast feature. This command prevents the interface from sending or receiving BPDUs.

⚠️

Caution   Enabling BPDU filtering on an interface is the same as disabling spanning tree on it and can result in spanning-tree loops.

You can enable the BPDU filtering feature for the entire switch or for an interface.

UplinkFast

Switches in hierarchical networks can be grouped into backbone switches, distribution switches, and access switches. Figure 22-2 shows a complex network where distribution switches and access switches each have at least one redundant link that spanning tree blocks to prevent loops.

Figure 22-2   Switches in a Hierarchical Network
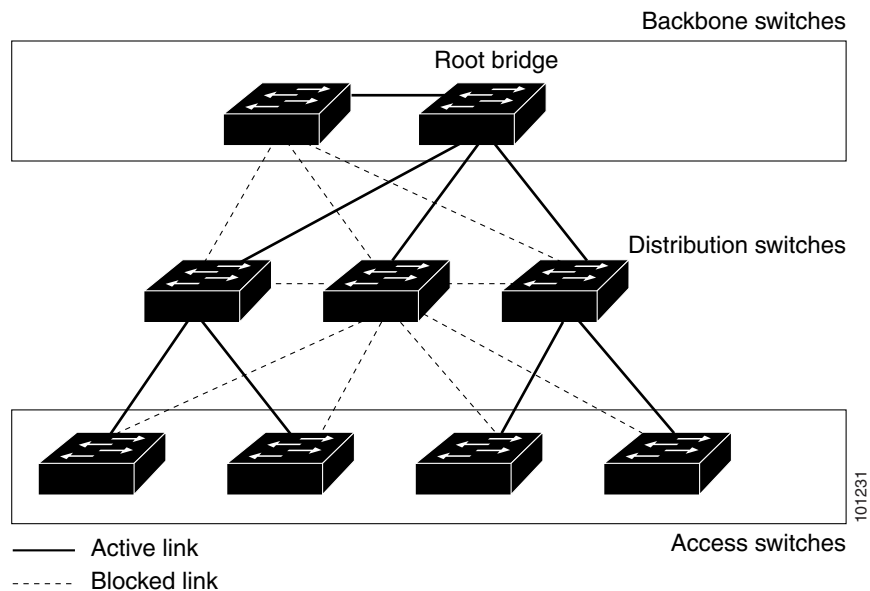
If a switch loses connectivity, it begins using the alternate paths as soon as the spanning tree selects a new root port. By enabling UplinkFast with the **spanning-tree uplinkfast** global configuration command, you can accelerate the choice of a new root port when a link or switch fails or when the spanning tree reconfigures itself. The root port transitions to the forwarding state immediately without going through the listening and learning states, as it would with the normal spanning-tree procedures.

When the spanning tree reconfigures the new root port, other interfaces flood the network with multicast packets, one for each address that was learned on the interface. You can limit these bursts of multicast traffic by reducing the max-update-rate parameter (the default for this parameter is 150 packets per second). However, if you enter zero, station-learning frames are not generated, so the spanning-tree topology converges more slowly after a loss of connectivity.

Note   UplinkFast is most useful in wiring-closet switches at the access or edge of the network. It is not appropriate for backbone devices. This feature might not be useful for other types of applications.

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 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.

Figure 22-3 shows an example topology with no link failures. Switch A, the root switch, is connected directly to Switch B over link L1 and to Switch C over link L2. The Layer 2 interface on Switch C that is connected directly to Switch B is in a blocking state.

Figure 22-3   UplinkFast Example Before Direct Link Failure
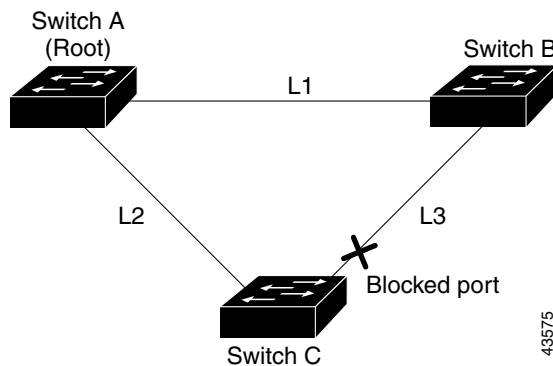
If Switch C detects a link failure on the currently active link L2 on the root port (a *direct* link failure), UplinkFast unblocks the blocked interface on Switch C and transitions it to the forwarding state without going through the listening and learning states, as shown in Figure 22-4. This change takes approximately 1 to 5 seconds.

Figure 22-4   UplinkFast Example After Direct Link Failure
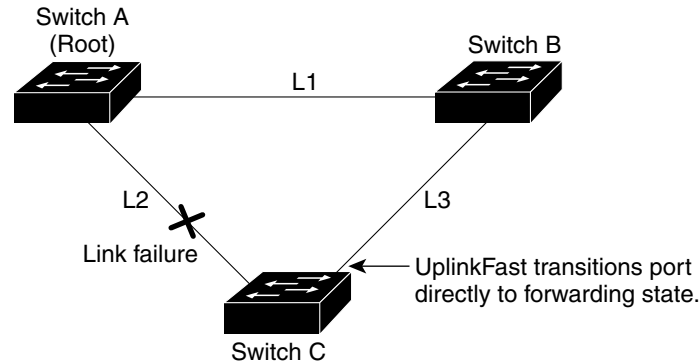
BackboneFast

BackboneFast detects indirect failures in the core of the backbone. BackboneFast is a complementary technology to the UplinkFast feature, which responds to failures on links directly connected to access switches. BackboneFast optimizes the maximum-age timer, which controls the amount of time the switch stores protocol information received on an interface. When a switch receives an inferior BPDU from the designated port of another switch, the BPDU is a signal that the other switch might have lost its path to the root, and BackboneFast tries to find an alternate path to the root.

BackboneFast, which is enabled by using the **spanning-tree backbonefast** global configuration command, starts when a root port or blocked interface on a switch receives inferior BPDUs from its designated switch. An inferior BPDU identifies a switch that declares itself as both the root bridge and the designated switch. When a switch receives an inferior BPDU, it means that a link to which the switch is not directly connected (an *indirect* link) has failed (that is, the designated switch has lost its connection to the root switch). Under spanning-tree rules, the switch ignores inferior BPDUs for the configured maximum aging time specified by the **spanning-tree vlan** *vlan-id max-age* global configuration command.

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

If the switch has alternate paths to the root switch, it uses these alternate paths to send a root link query (RLQ) request. The switch sends the RLQ request on all alternate paths and waits for an RLQ reply from other switches in the network.

If the switch discovers that it still has an alternate path to the root, it expires the maximum aging time on the interface that received the inferior BPDU. If all the alternate paths to the root switch indicate that the switch has lost connectivity to the root switch, the switch expires the maximum aging time on the interface that received the RLQ reply. If one or more alternate paths can still connect to the root switch, the switch makes all interfaces on which it received an inferior BPDU its designated ports and moves them from the blocking state (if they were in the blocking state), through the listening and learning states, and into the forwarding state.

Figure 22-5 shows an example topology with no link failures. Switch A, the root switch, connects directly to Switch B over link L1 and to Switch C over link L2. The Layer 2 interface on Switch C that connects directly to Switch B is in the blocking state.

Figure 22-5   ***BackboneFast Example Before Indirect Link Failure***
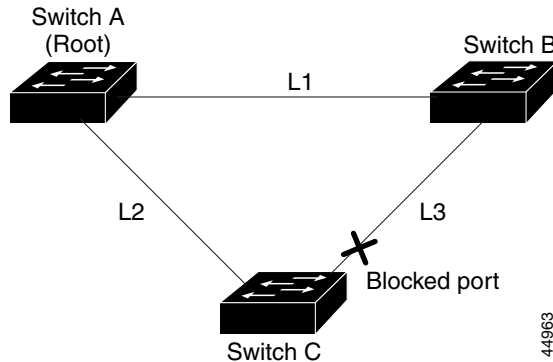

If link L1 fails as shown in Figure 22-6, Switch C cannot detect this failure because it is not connected directly to link L1. However, because Switch B is directly connected to the root switch over L1, it detects the failure, elects itself the root, and begins sending BPDUs to Switch C, identifying itself as the root. When Switch C receives the inferior BPDUs from Switch B, Switch C assumes that an indirect failure has occurred. At that point, BackboneFast allows the blocked interface on Switch C to move immediately to the listening state without waiting for the maximum aging time for the interface to expire. BackboneFast then transitions the Layer 2 interface on Switch C to the forwarding state, providing a path from Switch B to Switch A. The root-switch election takes approximately 30 seconds, twice the Forward Delay time if the default Forward Delay time of 15 seconds is set. Figure 22-6 shows how BackboneFast reconfigures the topology to account for the failure of link L1.

Figure 22-6   ***BackboneFast Example After Indirect Link Failure***
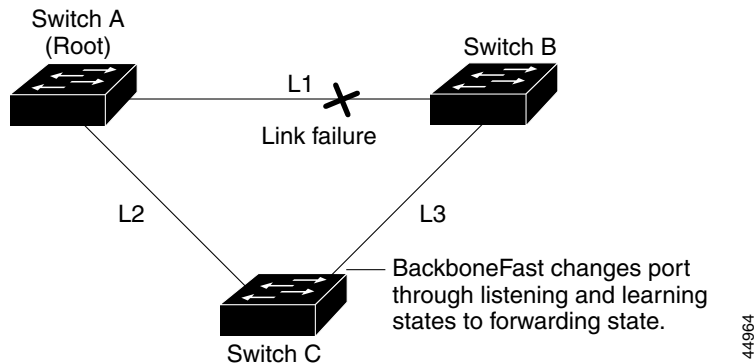
If a new switch is introduced into a shared-medium topology as shown in Figure 22-7, BackboneFast is not activated because the inferior BPDUs did not come from the recognized designated switch (Switch B). The new switch begins sending inferior BPDUs that indicate it is the root switch. However, the other switches ignore these inferior BPDUs, and the new switch learns that Switch B is the designated switch to Switch A, the root switch.

Figure 22-7   *Adding a Switch in a Shared-Medium Topology*
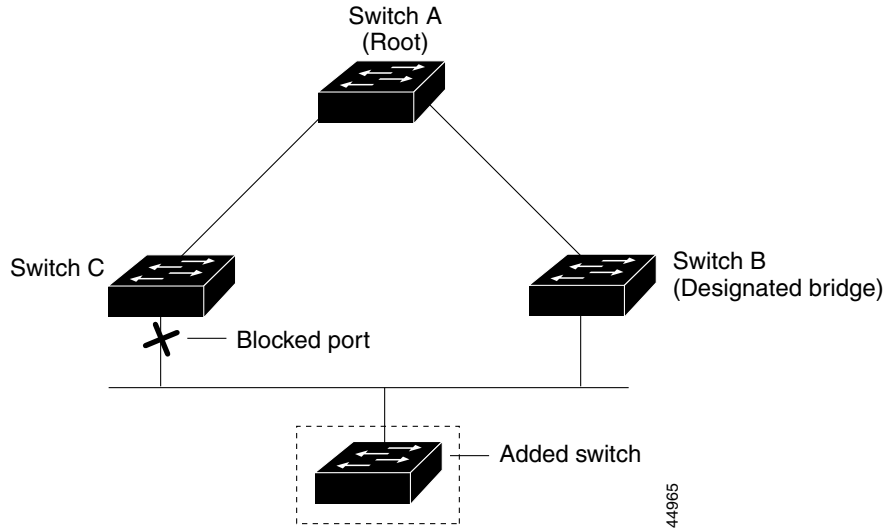
EtherChannel Guard

You can use EtherChannel guard to detect an EtherChannel misconfiguration between the switch and a connected device. A misconfiguration can occur if the switch interfaces are configured in an EtherChannel, but the interfaces on the other device are not. A misconfiguration can also occur if the channel parameters are not the same at both ends of the EtherChannel. For EtherChannel configuration guidelines, see the "EtherChannel Configuration Guidelines" section on page 40-10.

If the switch detects a misconfiguration on the other device, EtherChannel guard places the switch interfaces in the error-disabled state, and displays an error message.

You can enable this feature by using the **spanning-tree etherchannel guard misconfig** global configuration command.

Root Guard

The Layer 2 network of a service provider (SP) can include many connections to switches that are not owned by the SP. In such a topology, the spanning tree can reconfigure itself and select a *customer switch* as the root switch, as shown in Figure 22-8. You can avoid this situation by enabling root guard on SP switch interfaces that connect to switches in your customer's network. If spanning-tree calculations cause an interface in the customer network to be selected as the root port, root guard then places the interface in the root-inconsistent (blocked) state to prevent the customer's switch from becoming the root switch or being in the path to the root.

If a switch outside the SP network becomes the root switch, the interface is blocked (root-inconsistent state), and spanning tree selects a new root switch. The customer's switch does not become the root switch and is not in the path to the root.

If the switch is operating in multiple spanning-tree (MST) mode, root guard forces the interface to be a designated port. If a boundary port is blocked in an internal spanning-tree (IST) instance because of root guard, the interface also is blocked in all MST instances. A boundary port is an interface that connects to a LAN, the designated switch of which is either an IEEE 802.1D switch or a switch with a different MST region configuration.

Root guard enabled on an interface applies to all the VLANs to which the interface belongs. VLANs can be grouped and mapped to an MST instance.

You can enable this feature by using the **spanning-tree guard root** interface configuration command.

⚠️
Caution   Misuse of the root guard feature can cause a loss of connectivity.

Figure 22-8   Root Guard in a Service-Provider Network
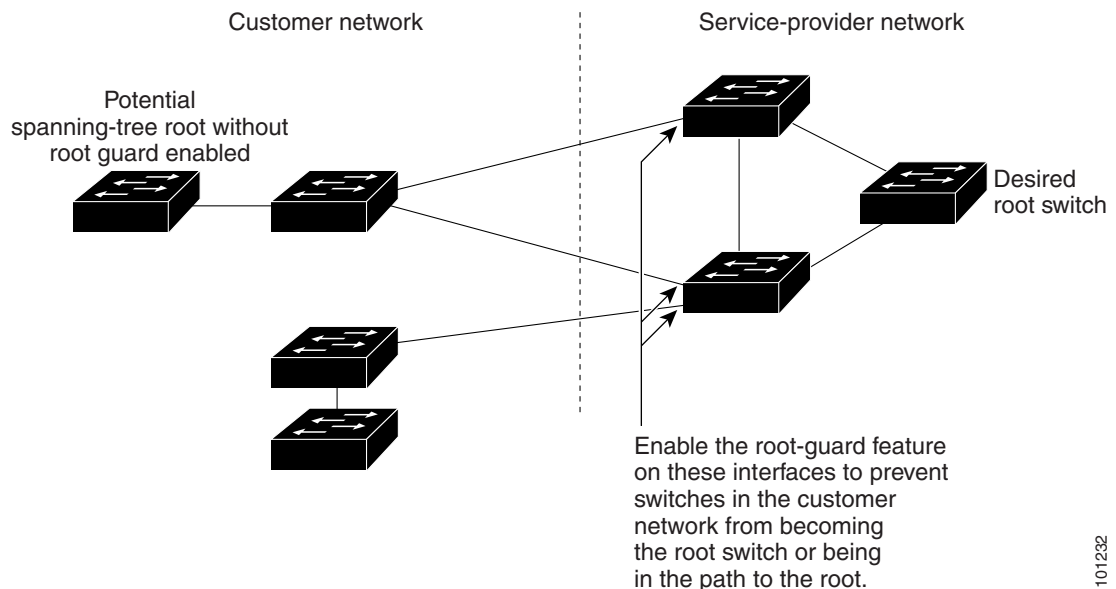
Loop Guard

You can use loop guard to prevent alternate or root ports from becoming designated ports because of a failure that leads to a unidirectional link. This feature is most effective when it is enabled on the entire switched network. Loop guard prevents alternate and root ports from becoming designated ports, and spanning tree does not send BPDUs on root or alternate ports.

You can enable this feature by using the **spanning-tree loopguard default** global configuration command.

When the switch is operating in PVST+ or rapid-PVST+ mode, loop guard prevents alternate and root ports from becoming designated ports, and spanning tree does not send BPDUs on root or alternate ports.

When the switch is operating in MST mode, BPDUs are not sent on nonboundary ports only if the interface is blocked by loop guard in all MST instances. On a boundary port, loop guard blocks the interface in all MST instances.

Default Optional Spanning-Tree Settings

Table 22-1   Default Optional Spanning-Tree Settings

Feature	Default Setting
PortFast, BPDU filtering, BPDU guard	Globally disabled (unless they are individually configured per interface).
UplinkFast	Globally disabled.
BackboneFast	Globally disabled.
EtherChannel guard	Globally enabled.
Root guard	Disabled on all interfaces.
Loop guard	Disabled on all interfaces.

How to Configure the Optional Spanning-Tree Features

Enabling Optional SPT Features

Before You Begin

- Make sure that there are no loops in the network between the trunk port and the workstation or server before you enable PortFast on a trunk port.

- Use PortFast *only* when connecting a single end station to an access or trunk port. Enabling this feature on an interface connected to a switch or hub could prevent spanning tree from detecting and disabling loops in your network, which could cause broadcast storms and address-learning problems.

- An interface with the PortFast feature enabled is moved directly to the spanning-tree forwarding state without waiting for the standard forward-time delay.

- You cannot enable both loop guard and root guard at the same time.

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

- If you enable the voice VLAN feature, the PortFast feature is automatically enabled. When you disable voice VLAN, the PortFast feature is not automatically disabled.

	Command	Purpose
Step 1	**show spanning-tree active** or **show spanning-tree mst**	Verifies which interfaces are alternate or root ports.
Step 2	**configure terminal**	Enters global configuration mode.
Step 3	**spanning-tree loopguard default**	Enables loop guard. By default, loop guard is disabled.

	Command	Purpose
Step 4	**spanning-tree portfast bpduguard default**	Enables BPDU guard.
		By default, BPDU guard is disabled.
Step 5	**spanning-tree portfast bpdufilter default**	Enables BPDU filtering.
		By default, BPDU filtering is disabled.
Step 6	**spanning-tree uplinkfast** [**max-update-rate** *pkts-per-second*]	Enables UplinkFast.
		(Optional) *pkts-per-second*—The range is 0 to 32000 packets per second; the default is 150.
		If you set the rate to 0, station-learning frames are not generated, and the spanning-tree topology converges more slowly after a loss of connectivity.
Step 7	**spanning-tree backbonefast**	Enables BackboneFast.
Step 8	**spanning-tree etherchannel guard misconfig**	Enables EtherChannel guard.
Step 9	**interface** *interface-id*	Specifies an interface to configure, and enters interface configuration mode.
Step 10	**spanning-tree portfast** [**trunk**]	Enables PortFast on an access port connected to a single workstation or server. By specifying the **trunk** keyword, you can enable PortFast on a trunk port.
		Note To enable PortFast on trunk ports, you must use the **spanning-tree portfast trunk** interface configuration command. The **spanning-tree portfast** command will not work on trunk ports.
		By default, PortFast is disabled on all interfaces.
Step 11	**spanning-tree guard root**	Enables root guard on the interface.
		By default, root guard is disabled on all interfaces.
Step 12	**end**	Returns to privileged EXEC mode.

Maintaining and Monitoring Optional Spanning-Tree Features

Command	Purpose
show spanning-tree active	Displays spanning-tree information on active interfaces only.
show spanning-tree detail	Displays a detailed summary of interface information.
show spanning-tree interface *interface-id*	Displays spanning-tree information for the specified interface.
show spanning-tree mst interface *interface-id*	Displays MST information for the specified interface.
show spanning-tree summary [**totals**]	Displays a summary of interface states or displays the total lines of the spanning-tree state section.
show interfaces status err-disabled	Displays which switch ports are disabled because of an EtherChannel misconfiguration.

Command	Purpose
show etherchannel summary	Displays the EtherChannel configuration. Useful to use on the remote device after switch ports are disabled.
[**no**] **shutdown**	Disables the interface. The **no** option reenables the interface.

Additional References

The following sections provide references related to switch administration:

Related Documents

Related Topic	Document Title
Cisco IE 2000 commands	*Cisco IE 2000 Switch Command Reference*, Release 15.0(1)EY
Cisco IOS basic commands	*Cisco IOS Configuration Fundamentals Command Reference*
VLAN configuration	*Chapter 17, "Configuring VLANs"*
Voice VLAN configuration	*Chapter 19, "Configuring Voice VLAN"*
PVST+ and rapid PVST+ configuratio	*Chapter 20, "Configuring STP"*
Multiple Spanning Tree Protocol configuration	*Chapter 21, "Configuring MSTP"*

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—