



Configuring Ethernet VLAN Trunks

This chapter describes how to configure Ethernet VLAN trunks on the Catalyst 6500 series switches.



Note

For complete information on configuring VLANs, see [Chapter 11, “Configuring VLANs.”](#)



Note

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

This chapter consists of these sections:

- [Understanding How VLAN Trunks Work, page 5-1](#)
- [Default Trunk Configuration, page 5-5](#)
- [Configuring a Trunk Link, page 5-6](#)
- [Example VLAN Trunk Configurations, page 5-12](#)
- [Disabling VLAN 1 on Trunks, page 5-10](#)
- [Enabling 802.1Q Tagging of Native VLAN Traffic, page 5-10](#)
- [Disabling 802.1Q Tagging on Specific Ports, page 5-11](#)

Understanding How VLAN Trunks Work

These sections describe how VLAN trunks work on the Catalyst 6500 series switches:

- [Trunking Overview, page 5-2](#)
- [Trunking Modes and Encapsulation Type, page 5-2](#)
- [802.1Q Trunk Configuration Guidelines and Restrictions, page 5-4](#)

Trunking Overview

A trunk is a point-to-point link between one or more Ethernet switch ports and another networking device such as a router or a switch. Trunks carry the traffic of multiple VLANs over a single link and allow you to extend VLANs across an entire network.

Two trunking encapsulations are available on all Ethernet ports:

- Inter-Switch Link (ISL)—ISL is a Cisco-proprietary trunking encapsulation
- IEEE 802.1Q—802.1Q is an industry-standard trunking encapsulation

You can configure a trunk on a single Ethernet port or on an EtherChannel bundle. For more information about EtherChannel, see [Chapter 6, “Configuring EtherChannel.”](#)

Ethernet trunk ports support five different trunking modes (see [Table 5-1](#)). In addition, you can specify whether the trunk will use ISL encapsulation, 802.1Q encapsulation, or whether the encapsulation type will be autonegotiated.

For trunking to be autonegotiated, the ports must be in the same VLAN Trunking Protocol (VTP) domain. However, you can use the **on** or **nonegotiate** mode to force a port to become a trunk, even if it is in a different domain. For more information on VTP domains, see [Chapter 10, “Configuring VTP.”](#)

Trunk negotiation is managed by the Dynamic Trunking Protocol (DTP). DTP supports autonegotiation of both ISL and 802.1Q trunks.

Trunking Modes and Encapsulation Type



Note

The following 10-Gigabit Ethernet Switching Modules do not support ISL encapsulation: WS-X6501-10GEX4 and WS-X6502-10GE.

[Table 5-1](#) lists the trunking modes used with the **set trunk** command and describes how they function on Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet ports.

Table 5-1 Ethernet Trunking Modes

Mode	Function
on	Puts the port into permanent trunking mode and negotiates to convert the link into a trunk link. The port becomes a trunk port even if the neighboring port does not agree to the change.
off	Puts the port into permanent nontrunking mode and negotiates to convert the link into a nontrunk link. The port becomes a nontrunk port even if the neighboring port does not agree to the change.
desirable	Makes the port actively attempt to convert the link to a trunk link. The port becomes a trunk port if the neighboring port is set to on , desirable , or auto mode.
auto	Makes the port willing to convert the link to a trunk link. The port becomes a trunk port if the neighboring port is set to on or desirable mode. This is the default mode for all Ethernet ports.
nonegotiate	Puts the port into permanent trunking mode but prevents the port from generating DTP frames. You must configure the neighboring port manually as a trunk port to establish a trunk link.

[Table 5-2](#) lists the encapsulation types that are used with the **set trunk** command and describes how they function on Ethernet ports. You can use the **show port capabilities** command to determine which encapsulation types a particular port supports.

Table 5-2 Ethernet Trunk Encapsulation Types

Encapsulation	Function
isl	Specifies ISL encapsulation on the trunk link.
dot1q	Specifies 802.1Q encapsulation on the trunk link.
negotiate	Specifies that the port negotiate with the neighboring port to become an ISL (preferred) or 802.1Q trunk, depending on the configuration and capabilities of the neighboring port.

The trunking mode, the trunk encapsulation type, and the hardware capabilities of the two connected ports determine whether a trunk link comes up and the type of trunk the link becomes. Table 5-3 shows the result of the possible trunking configurations.

Table 5-3 Results of Possible Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet Trunk Configurations

Neighbor Port Trunk Mode and Trunk Encapsulation	Local Port Trunk Mode and Trunk Encapsulation								
	off isl or dot1q	on isl	desirable isl	auto isl	on dot1q	desirable dot1q	auto dot1q	desirable negotiate	auto negotiate
off isl or dot1q	Local: Nontrunk	Local: ISL trunk	Local: Nontrunk	Local: Nontrunk	Local: 1Q trunk	Local: Nontrunk	Local: Nontrunk	Local: Nontrunk	Local: Nontrunk
	Neighbor: Nontrunk	Neighbor: Nontrunk	Neighbor: Nontrunk	Neighbor: Nontrunk	Neighbor: Nontrunk	Neighbor: Nontrunk	Neighbor: Nontrunk	Neighbor: Nontrunk	Neighbor: Nontrunk
on isl	Local: Nontrunk	Local: ISL trunk	Local: ISL trunk	Local: ISL trunk	Local: 1Q trunk ¹	Local: Nontrunk	Local: Nontrunk	Local: ISL trunk	Local: ISL trunk
	Neighbor: ISL trunk	Neighbor: ISL trunk	Neighbor: ISL trunk	Neighbor: ISL trunk	Neighbor: ISL trunk ¹	Neighbor: ISL trunk	Neighbor: ISL trunk	Neighbor: ISL trunk	Neighbor: ISL trunk
desirable isl	Local: Nontrunk	Local: ISL trunk	Local: ISL trunk	Local: ISL trunk	Local: 1Q trunk	Local: Nontrunk	Local: Nontrunk	Local: ISL trunk	Local: ISL trunk
	Neighbor: Nontrunk	Neighbor: ISL trunk	Neighbor: ISL trunk	Neighbor: ISL trunk	Neighbor: Nontrunk	Neighbor: Nontrunk	Neighbor: Nontrunk	Neighbor: ISL trunk	Neighbor: ISL trunk
auto isl	Local: Nontrunk	Local: ISL trunk	Local: ISL trunk	Local: Nontrunk	Local: 1Q trunk	Local: Nontrunk	Local: Nontrunk	Local: ISL trunk	Local: Nontrunk
	Neighbor: Nontrunk	Neighbor: ISL trunk	Neighbor: ISL trunk	Neighbor: Nontrunk	Neighbor: Nontrunk	Neighbor: Nontrunk	Neighbor: Nontrunk	Neighbor: ISL trunk	Neighbor: Nontrunk
on dot1q	Local: Nontrunk	Local: ISL trunk ¹	Local: Nontrunk	Local: Nontrunk	Local: 1Q trunk	Local: 1Q trunk	Local: 1Q trunk	Local: 1Q trunk	Local: 1Q trunk
	Neighbor: 1Q trunk	Neighbor: 1Q trunk ¹	Neighbor: 1Q trunk	Neighbor: 1Q trunk	Neighbor: 1Q trunk	Neighbor: 1Q trunk	Neighbor: 1Q trunk	Neighbor: 1Q trunk	Neighbor: 1Q trunk
desirable dot1q	Local: Nontrunk	Local: ISL trunk	Local: Nontrunk	Local: Nontrunk	Local: 1Q trunk	Local: 1Q trunk	Local: 1Q trunk	Local: 1Q trunk	Local: 1Q trunk
	Neighbor: Nontrunk	Neighbor: Nontrunk	Neighbor: Nontrunk	Neighbor: Nontrunk	Neighbor: 1Q trunk	Neighbor: 1Q trunk	Neighbor: 1Q trunk	Neighbor: 1Q trunk	Neighbor: 1Q trunk

Table 5-3 Results of Possible Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet Trunk Configurations (continued)

Neighbor Port Trunk Mode and Trunk Encapsulation	Local Port Trunk Mode and Trunk Encapsulation								
	off isl or dot1q	on isl	desirable isl	auto isl	on dot1q	desirable dot1q	auto dot1q	desirable negotiate	auto negotiate
auto dot1q	Local: Nontrunk	Local: ISL trunk	Local: Nontrunk	Local: Nontrunk	Local: 1Q trunk	Local: 1Q trunk	Local: Nontrunk	Local: 1Q trunk	Local: Nontrunk
	Neighbor: Nontrunk	Neighbor: Nontrunk	Neighbor: Nontrunk	Neighbor: Nontrunk	Neighbor: 1Q trunk	Neighbor: 1Q trunk	Neighbor: Nontrunk	Neighbor: 1Q trunk	Neighbor: Nontrunk
desirable negotiate	Local: Nontrunk	Local: ISL trunk	Local: ISL trunk	Local: ISL trunk	Local: 1Q trunk	Local: 1Q trunk	Local: 1Q trunk	Local: ISL trunk	Local: ISL trunk
	Neighbor: Nontrunk	Neighbor: ISL trunk	Neighbor: ISL trunk	Neighbor: ISL trunk	Neighbor: 1Q trunk	Neighbor: 1Q trunk	Neighbor: 1Q trunk	Neighbor: ISL trunk	Neighbor: ISL trunk
auto negotiate	Local: Nontrunk	Local: ISL trunk	Local: ISL trunk	Local: Nontrunk	Local: 1Q trunk	Local: 1Q trunk	Local: Nontrunk	Local: ISL trunk	Local: Nontrunk
	Neighbor: Nontrunk	Neighbor: ISL trunk	Neighbor: ISL trunk	Neighbor: Nontrunk	Neighbor: 1Q trunk	Neighbor: 1Q trunk	Neighbor: Nontrunk	Neighbor: ISL trunk	Neighbor: Nontrunk

- Using this configuration can result in spanning tree loops and is not recommended.

**Note**

DTP is a point-to-point protocol. However, some internetworking devices might forward DTP frames improperly. To avoid this problem, ensure that trunking is turned **off** on ports that are connected to non-switch devices if you do not intend to trunk across those links. When manually enabling trunking on a link to a Cisco router, use the **nonnegotiate** keyword to cause the port to become a trunk but not generate DTP frames.

802.1Q Trunk Configuration Guidelines and Restrictions

The following configuration guidelines and restrictions apply when using 802.1Q trunks and impose some limitations on the trunking strategy for a network:

- When connecting Cisco switches through an 802.1Q trunk, make sure that the native VLAN for an 802.1Q trunk is the same on both ends of the trunk link. If the native VLAN on one end of the trunk is different from the native VLAN on the other end, spanning tree loops might result.
- Disabling spanning tree on the native VLAN of an 802.1Q trunk without disabling spanning tree on every VLAN in the network can cause spanning tree loops. We recommend that you leave spanning tree enabled on the native VLAN of an 802.1Q trunk. If this is not possible, disable spanning tree on every VLAN in the network. Make sure that your network is free of physical loops before disabling spanning tree.
- When you connect two Cisco switches through 802.1Q trunks, the switches exchange spanning tree BPDUs on each VLAN that is allowed on the trunks. The BPDUs on the native VLAN of the trunk are sent untagged to the reserved IEEE 802.1D spanning tree multicast MAC address (01-80-C2-00-00-00). The BPDUs on all other VLANs on the trunk are sent tagged to the reserved Cisco Shared Spanning Tree (SSTP) multicast MAC address (01-00-0c-cc-cc-cd).

- Non-Cisco 802.1Q switches maintain only a single instance of spanning tree (the Mono Spanning Tree, or MST) that defines the spanning-tree topology for all VLANs. When you connect a Cisco switch to a non-Cisco switch through an 802.1Q trunk, the MST of the non-Cisco switch and the native VLAN spanning-tree of the Cisco switch combine to form a single spanning-tree topology known as the Common Spanning Tree (CST).

When you connect a Cisco switch to a non-Cisco switch, the CST is always on VLAN 1. The Cisco switch sends an untagged IEEE BPDU (01-80-C2-00-00-00) on VLAN 1 for the CST. On the native VLAN, the Cisco switch sends an untagged Cisco BPDU (01-00-0C-CC-CC-CC) which the non-Cisco switch forwards but does not act on (the IEEE BPDU is not forwarded on the native VLAN).

- Because Cisco switches transmit BPDUs to the SSTP multicast MAC address on VLANs other than the native VLAN of the trunk, non-Cisco switches do not recognize these frames as BPDUs and flood them on all ports in the corresponding VLAN. Other Cisco switches that are connected to the non-Cisco 802.1Q cloud receive these flooded BPDUs. This allows Cisco switches to maintain a per-VLAN spanning tree topology across a cloud of non-Cisco 802.1Q switches. The non-Cisco 802.1Q cloud separating the Cisco switches is treated as a single broadcast segment between all switches connected to the non-Cisco 802.1Q cloud through 802.1Q trunks.
- Make certain that the native VLAN is the same on ALL of the 802.1Q trunks connecting the Cisco switches to the non-Cisco 802.1Q cloud.
- If you are connecting multiple Cisco switches to a non-Cisco 802.1Q cloud, all of the connections MUST be through 802.1Q trunks. You CANNOT connect Cisco switches to a non-Cisco 802.1Q cloud through ISL trunks or through access ports. Doing so will cause the switch to place the ISL trunk port or access port into the spanning tree “port inconsistent” state and no traffic will pass through the port.

Default Trunk Configuration

Table 5-4 shows the default Ethernet trunk configuration.

Table 5-4 *Default Ethernet Trunk Configuration*

Feature	Default Configuration
Trunk mode	auto
Trunk encapsulation	negotiate
Allowed VLAN range	VLANs 1–1005, 1025–4094

Configuring a Trunk Link

These sections describe how to configure a trunk link on Ethernet ports and how to define the allowed VLAN range on a trunk:

- [Configuring an ISL Trunk, page 5-6](#)
- [Configuring an 802.1Q Trunk, page 5-7](#)
- [Configuring an ISL/802.1Q Negotiating Trunk Port, page 5-8](#)
- [Defining the Allowed VLANs on a Trunk, page 5-8](#)
- [Disabling a Trunk Port, page 5-9](#)
- [Disabling VLAN 1 on Trunks, page 5-10](#)
- [Enabling 802.1Q Tagging of Native VLAN Traffic, page 5-10](#)
- [Disabling 802.1Q Tagging on Specific Ports, page 5-11](#)

Configuring an ISL Trunk

To configure an ISL trunk, perform this task in privileged mode:

	Task	Command
Step 1	Configure an ISL trunk.	set trunk <i>mod/port</i> [on desirable auto nonegotiate] isl
Step 2	Verify the trunking configuration.	show trunk [<i>mod/port</i>]

This example shows how to configure a port as a trunk and how to verify the trunk configuration. This example assumes that the neighboring port is in **auto** mode:

```

Console> (enable) set trunk 1/1 on
Port(s) 1/1 trunk mode set to on.
Console> (enable) 06/16/1998,22:16:39:DTP-5:Port 1/1 has become isl trunk
06/16/1998,22:16:40:PAGP-5:Port 1/1 left bridge port 1/1.
06/16/1998,22:16:40:PAGP-5:Port 1/1 joined bridge port 1/1.
Console> (enable) show trunk
Port      Mode      Encapsulation  Status      Native vlan
-----  -
1/1      on        isl             trunking    1
Port      Vlans allowed on trunk
-----  -
1/1      1-1005, 1025-4094

Port      Vlans allowed and active in management domain
-----  -
1/1      1,521-524
Port      Vlans in spanning tree forwarding state and not pruned
-----  -
1/1
Console> (enable)

```

This example shows how to place a port in **desirable** mode and how to verify the trunk configuration. This example assumes that the neighboring port is in **auto** mode:

```

Console> (enable) set trunk 1/2 desirable
Port(s) 1/2 trunk mode set to desirable.
Console> (enable) 06/16/1998,22:20:16:DTP-5:Port 1/2 has become isl trunk
06/16/1998,22:20:16:PAGP-5:Port 1/2 left bridge port 1/2.
06/16/1998,22:20:16:PAGP-5:Port 1/2 joined bridge port 1/2.
Console> (enable) show trunk 1/2
Port      Mode           Encapsulation  Status      Native vlan
-----
1/2      desirable     isl            trunking    1
Port      Vlans allowed on trunk
-----
1/2      1-1005, 1025-4094
Port      Vlans allowed and active in management domain
-----
1/2      1,521-524
Port      Vlans in spanning tree forwarding state and not pruned
-----
1/2
Console> (enable)

```

Configuring an 802.1Q Trunk

To configure an 802.1Q trunk, perform this task in privileged mode:

	Task	Command
Step 1	Configure an 802.1Q trunk.	set trunk <i>mod/port</i> [on desirable auto nonegotiate] dot1q
Step 2	Verify the trunking configuration.	show trunk [<i>mod/port</i>]

This example shows how to configure an 802.1Q trunk and how to verify the trunk configuration:

```

Console> (enable) set trunk 2/9 desirable dot1q
Port(s) 2/9 trunk mode set to desirable.
Port(s) 2/9 trunk type set to dot1q.
Console> (enable) 07/02/1998,18:22:25:DTP-5:Port 2/9 has become dot1q trunk

Console> (enable) show trunk
Port      Mode           Encapsulation  Status      Native vlan
-----
2/9      desirable     dot1q          trunking    1
Port      Vlans allowed on trunk
-----
2/9      1-1005, 1025-4094
Port      Vlans allowed and active in management domain
-----
2/9      1,5,10-32,101-120,150,200,250,300,400,500,600,700,800,900,1000
Port      Vlans in spanning tree forwarding state and not pruned
-----
2/9      5,10-32,101-120,150,200,250,300,400,500,600,700,800,900,1000
Console> (enable)

```

Configuring an ISL/802.1Q Negotiating Trunk Port

To configure a trunk port to negotiate the trunk encapsulation type (either ISL or 802.1Q), perform this task in privileged mode:

	Task	Command
Step 1	Configure a port to negotiate the trunk encapsulation type.	set trunk <i>mod/port</i> [on desirable auto nonegotiate] negotiate
Step 2	Verify the trunking configuration.	show trunk [<i>mod/port</i>]

This example shows how to configure a port to negotiate the encapsulation type and how to verify the trunk configuration. This example assumes that the neighboring port is in **auto** mode with encapsulation set to **isl** or **negotiate**.

```

Console> (enable) set trunk 4/11 desirable negotiate
Port(s) 4/11 trunk mode set to desirable.
Port(s) 4/11 trunk type set to negotiate.
Console> (enable) show trunk 4/11
Port      Mode           Encapsulation  Status      Native vlan
-----
4/11     desirable     n-isl          trunking    1

Port      Vlans allowed on trunk
-----
4/11     1-1005,1025-4094

Port      Vlans allowed and active in management domain
-----
4/11     1,5,10-32,55,101-120,998-1000

Port      Vlans in spanning tree forwarding state and not pruned
-----
4/11     1,5,10-32,55,101-120,998-1000
Console> (enable)

```

Defining the Allowed VLANs on a Trunk

When you configure a trunk port, all VLANs are added to the allowed VLANs list for that trunk. However, you can remove VLANs from the allowed list to prevent traffic for those VLANs from passing over the trunk.



Note

When you first configure a port as a trunk, entering the **set trunk** command always adds all VLANs to the allowed VLAN list for the trunk, even if you specify a VLAN range (any specified VLAN range is ignored). To modify the allowed VLANs list, use a combination of the **clear trunk** and **set trunk** commands to specify the allowed VLANs.

To define the allowed VLAN list for a trunk port, perform this task in privileged mode:

	Task	Command
Step 1	Remove VLANs from the allowed VLANs list for a trunk.	clear trunk <i>mod/port vlans</i>
Step 2	(Optional) Add specific VLANs to the allowed VLANs list for a trunk.	set trunk <i>mod/port vlans</i>
Step 3	Verify the allowed VLAN list for the trunk.	show trunk [<i>mod/port</i>]

This example shows how to define the allowed VLANs list to allow VLANs 1–100, VLANs 500–1005, and VLAN 2500 on trunk port 1/1 and how to verify the allowed VLAN list for the trunk:

```

Console> (enable) clear trunk 1/1 101-499
Removing Vlan(s) 101-499 from allowed list.
Port 1/1 allowed vlans modified to 1-100,500-1005.
Console> (enable) set trunk 1/1 2500
Adding vlans 2500 to allowed list.
Port(s) 1/1 allowed vlans modified to 1-100,500-1005,2500.
Console> (enable) show trunk 1/1
Port      Mode           Encapsulation  Status      Native vlan
-----
1/1      desirable     isl            trunking    1
Port      Vlans allowed on trunk
-----
1/1      1-100, 500-1005,2500
Port      Vlans allowed and active in management domain
-----
1/1      1,521-524
Port      Vlans in spanning tree forwarding state and not pruned
-----
1/1      1,521-524
Console> (enable)

```

Disabling a Trunk Port

To turn off trunking on a port, perform this task in privileged mode:

	Task	Command
Step 1	Turn off trunking on a port.	set trunk <i>mod/port off</i>
Step 2	Verify the trunking configuration.	show trunk [<i>mod/port</i>]

To return a port to the default trunk type and mode for that port type, perform this task in privileged mode:

	Task	Command
Step 1	Return the port to the default trunking type and mode for that port type.	clear trunk <i>mod/port</i>
Step 2	Verify the trunking configuration.	show trunk [<i>mod/port</i>]

Disabling VLAN 1 on Trunks

On the Catalyst 6500 series switches, VLAN 1 is enabled by default to allow control protocols to transmit and receive packets across the network topology. However, when VLAN 1 is enabled on trunk links in a large complex network, the impact of broadcast storms increases. Because spanning tree applies to the entire network, spanning tree loops might increase when you enable VLAN 1 on all trunk links. To prevent this scenario, you can disable VLAN 1 on trunk interfaces.

When you disable VLAN 1 on a trunk interface, no user traffic is transmitted and received across that trunk interface, but the supervisor engine continues to transmit and receive packets from control protocols such as Cisco Discovery Protocol (CDP), VTP, Port Aggregation Protocol (PAgP), and DTP.

When a trunk port with VLAN 1 disabled becomes a nontrunk port, it is added to the native VLAN. If the native VLAN is VLAN 1, the port is enabled and added to VLAN 1.

To disable VLAN 1 on a trunk interface, perform this task in privileged mode:

	Task	Command
Step 1	Disable VLAN 1 on the trunk interface.	clear trunk <i>mod/port</i> [<i>vlan-range</i>]
Step 2	Verify the allowed VLAN list for the trunk.	show trunk [<i>mod/port</i>]

This example shows how to disable VLAN 1 on a trunk link and verify the configuration:

```

Console> (enable) clear trunk 8/1 1
Removing Vlan(s) 1 from allowed list.
Port 8/1 allowed vlans modified to 2-1005.
Console> (enable) show trunk 8/1
Port      Mode           Encapsulation  Status        Native vlan
-----
8/1      on             isl            trunking      1

Port      Vlans allowed on trunk
-----
8/1      2-1005, 1025-4094

Port      Vlans allowed and active in management domain
-----
8/1      2-6,10,20,50,100,152,200,300,400,500,521,524,570,776,801-802,850,917,999,1003,1005

Port      Vlans in spanning tree forwarding state and not pruned
-----
8/1      2-6,10,20,50,100,152,200,300,400,500,521,524,570,776,802,850,917,999,1003,1005
Console> (enable) show config

```

Enabling 802.1Q Tagging of Native VLAN Traffic

The **set dot1q-all-tagged enable** command is a global command that configures a switch to forward all frames from 802.1Q trunks with 802.1Q tagging in the native VLAN and admit only 802.1Q tagged frames on 802.1Q trunks, dropping any untagged traffic, including untagged traffic in the native VLAN. You can enter this command on any switch that needs to support 802.1Q tunneling with 802.1Q trunks.

To configure the switch to forward all 802.1Q tagged frames on 802.1Q trunks, perform this task in privileged mode:

	Task	Command
Step 1	Enable the switch to forward all 802.1Q tagged frames.	set dot1q-all-tagged [enable disable]
Step 2	Verify the configuration.	show dot1q-all-tagged

This example shows how to enable the switch to forward all 802.1Q traffic and verify the configuration:

```
Console> (enable) set dot1q-all-tagged enable
Dot1q-all-tagged feature enabled globally.
Console> (enable) show dot1q-all-tagged
Dot1q-all-tagged feature globally enabled.
Console> (enable)
```

Disabling 802.1Q Tagging on Specific Ports

The **set port dot1q-all-tagged mod/port enable | disable** command allows you to disable 802.1Q tagging on specific ports. Enter the **set port dot1q-all-tagged disable** command to selectively disable 802.1Q tagging on ports that connect to the devices that do not support 802.1Q tagged traffic. If you enable or disable 802.1Q tagging on an EtherChannel port, the configuration is applied to all ports in the channel.



Note

If you did not enter the global **set dot1q-all-tagged enable** command, the default group is never tagged and the per-port setting has no effect.

If you entered the global **set dot1q-all-tagged enable** command, the per-port setting controls whether frames are tagged.



Note

The **set port dot1q-all-tagged mod/port enable | disable** command is not supported on the ports on the MSFC or ports on the WS-X6101 OC-12 ATM modules.

To disable the forwarding of 802.1Q tagged frames on specific ports, perform this task in privileged mode:

	Task	Command
Step 1	Enable or disable the forwarding of 802.1Q tagged frames on specific ports or on all ports.	set port dot1q-all-tagged mod/port enable disable
Step 2	Verify the configuration.	show port dot1q-all-tagged

This example shows how to disable the forwarding of 802.1Q tagged frames on port 3/2 and verify the configuration:

```
Console> (enable) set port dot1q-all-tagged 3/2 disable
Packets on native vlan will not be tagged on port 3/2.
Console> (enable) show port dot1q-all-tagged
```

```
Dot1q-all-tagged feature globally enabled.
Port      Dot1q-all-tagged mode
----      -
2/1      enable
2/2      enable
3/1      enable
3/2      disable
3/3      enable
3/4      enable
3/5      enable
<output truncated>
```

Example VLAN Trunk Configurations

This section contains example VLAN trunk configurations:

- [ISL Trunk Configuration Example, page 5-12](#)
- [ISL Trunk Over EtherChannel Link Example, page 5-13](#)
- [802.1Q Trunk Over EtherChannel Link Example, page 5-16](#)
- [Load-Sharing VLAN Traffic Over Parallel Trunks Example, page 5-20](#)

ISL Trunk Configuration Example

This example shows how to configure an ISL trunk between two switches and how to limit the allowed VLANs on the trunk to VLAN 1 and VLANs 520–530.

In this example, port 1/1 on Switch 1 is connected to a Fast Ethernet port on another switch. Both ports are in their default state, with the trunk mode set to **auto** (for more information, see the “[Default Trunk Configuration](#)” section on page 5-5).

-
- Step 1** Configure port 1/1 on Switch 1 as an ISL trunk port by entering the **set trunk** command. By specifying the **desirable** keyword, the trunk is automatically negotiated with the neighboring port (port 1/2 on Switch 2). ISL encapsulation is assumed based on the hardware type.

```
Switch1> (enable) set trunk 1/1 desirable
Port(s) 1/1 trunk mode set to desirable.
Switch1> (enable) 06/18/1998,12:20:23:DTP-5:Port 1/1 has become isl trunk
06/18/1998,12:20:23:PAGP-5:Port 1/1 left bridge port 1/1.
06/18/1998,12:20:23:PAGP-5:Port 1/1 joined bridge port 1/1.
Switch1> (enable)
```

- Step 2** Check the configuration by entering the **show trunk** command. The Status field in the screen output indicates that port 1/1 is trunking.

```
Switch1> (enable) show trunk 1/1
Port      Mode           Encapsulation  Status        Native vlan
-----
1/1      desirable     isl            trunking      1
Port      Vlans allowed on trunk
-----
1/1      1-1005, 1025-4094
Port      Vlans allowed and active in management domain
-----
1/1      1,521-524
Port      Vlans in spanning tree forwarding state and not pruned
-----
1/1
Switch1> (enable)
```

- Step 3** Define the allowed VLAN list for the trunk by entering the **clear trunk** command to remove the VLANs that should not pass traffic over the trunk link.

```
Switch1> (enable) clear trunk 1/1 2-519
Removing Vlan(s) 2-519 from allowed list.
Port 1/1 allowed vlans modified to 1,520-1005.
Switch1> (enable) clear trunk 1/1 531-1005
Removing Vlan(s) 531-1005 from allowed list.
Port 1/1 allowed vlans modified to 1,520-530.
Switch1> (enable) show trunk 1/1
Port      Mode           Encapsulation  Status        Native vlan
-----
1/1      desirable     isl            trunking      1
Port      Vlans allowed on trunk
-----
1/1      1,520-530
Port      Vlans allowed and active in management domain
-----
1/1      1,521-524
Port      Vlans in spanning tree forwarding state and not pruned
-----
1/1      1,521-524
Switch1> (enable)
```

- Step 4** Verify connectivity across the trunk by entering the **ping** command.

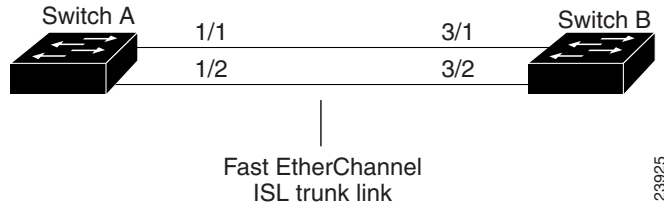
```
Switch1> (enable) ping switch2
switch2 is alive
Switch1> (enable)
```

ISL Trunk Over EtherChannel Link Example

This example shows how to configure an ISL trunk over an EtherChannel link between two switches.

[Figure 5-1](#) shows two switches that are connected through two 100BASE-TX Fast Ethernet ports.

Figure 5-1 ISL Trunk Over Fast EtherChannel Link



This example shows how to configure the switches to form a two-port EtherChannel bundle and then configure the EtherChannel bundle as an ISL trunk link.

- Step 1** Confirm the channeling and trunking status of the switches by entering the **show port channel** and **show trunk** commands.

```
Switch_A> (enable) show port channel
No ports channelling
Switch_A> (enable) show trunk
No ports trunking.
Switch_A> (enable)
```

```
Switch_B> (enable) show port channel
No ports channelling
Switch_B> (enable) show trunk
No ports trunking.
Switch_B> (enable)
```

- Step 2** Configure the ports on Switch A to negotiate an EtherChannel bundle with the neighboring switch by entering the **set port channel** command. This example assumes that the neighboring ports on Switch B are in EtherChannel **auto** mode. The system logging messages provide information about the formation of the EtherChannel bundle.

```
Switch_A> (enable) set port channel 1/1-2 desirable
Port(s) 1/1-2 channel mode set to desirable.
Switch_A> (enable) %PAGP-5-PORTFROMSTP:Port 1/1 left bridge port 1/1
%PAGP-5-PORTFROMSTP:Port 1/2 left bridge port 1/2
%PAGP-5-PORTFROMSTP:Port 1/2 left bridge port 1/2
%PAGP-5-PORTTOSTP:Port 1/1 joined bridge port 1/1-2
%PAGP-5-PORTTOSTP:Port 1/2 joined bridge port 1/1-2

Switch_B> (enable) %PAGP-5-PORTFROMSTP:Port 3/1 left bridge port 3/1
%PAGP-5-PORTFROMSTP:Port 3/2 left bridge port 3/2
%PAGP-5-PORTFROMSTP:Port 3/2 left bridge port 3/2
%PAGP-5-PORTTOSTP:Port 3/1 joined bridge port 3/1-2
%PAGP-5-PORTTOSTP:Port 3/2 joined bridge port 3/1-2
```

- Step 3** After the EtherChannel bundle is negotiated, verify the configuration by entering the **show port channel** command.

```
Switch_A> (enable) show port channel
Port  Status      Channel  Channel  Neighbor  Neighbor
      mode        status   device   device
-----
1/1   connected    desirable channel   WS-C5000  009979082(Sw 3/1
1/2   connected    desirable channel   WS-C5000  009979082(Sw 3/2
-----
Switch_A> (enable)
```

```
Switch_B> (enable) show port channel
Port  Status      Channel  Channel  Neighbor  Neighbor
      mode        status   device   device
-----
3/1   connected    auto     channel   WS-C5500  069003103(Sw 1/1
3/2   connected    auto     channel   WS-C5500  069003103(Sw 1/2
-----
Switch_B> (enable)
```

- Step 4** Configure one of the ports in the EtherChannel bundle to negotiate an ISL trunk by entering the **set trunk** command.

The configuration is applied to all of the ports in the bundle. This example assumes that the neighboring ports on Switch B are configured to use **isl** or **negotiate** encapsulation and are in **auto** trunk mode. The system logging messages provide information about the formation of the ISL trunk.

```
Switch_A> (enable) set trunk 1/1 desirable isl
Port(s) 1/1-2 trunk mode set to desirable.
Port(s) 1/1-2 trunk type set to isl.
Switch_A> (enable) %DTP-5-TRUNKPORTON:Port 1/1 has become isl trunk
%DTP-5-TRUNKPORTON:Port 1/2 has become isl trunk
%PAGP-5-PORTFROMSTP:Port 1/1 left bridge port 1/1-2
%PAGP-5-PORTFROMSTP:Port 1/2 left bridge port 1/1-2
%PAGP-5-PORTTOSTP:Port 1/1 joined bridge port 1/1-2
%PAGP-5-PORTTOSTP:Port 1/2 joined bridge port 1/1-2

Switch_B> (enable) %DTP-5-TRUNKPORTON:Port 3/1 has become isl trunk
%DTP-5-TRUNKPORTON:Port 3/2 has become isl trunk
%PAGP-5-PORTFROMSTP:Port 3/1 left bridge port 3/1-2
%PAGP-5-PORTFROMSTP:Port 3/2 left bridge port 3/1-2
%PAGP-5-PORTTOSTP:Port 3/1 joined bridge port 3/1-2
%PAGP-5-PORTTOSTP:Port 3/2 joined bridge port 3/1-2
```

- Step 5** After the ISL trunk link is negotiated, verify the configuration by entering the **show trunk** command.

```
Switch_A> (enable) show trunk
Port      Mode          Encapsulation  Status      Native vlan
-----
1/1      desirable    isl             trunking    1
1/2      desirable    isl             trunking    1

Port      Vlans allowed on trunk
-----
1/1      1-1005, 1025-4094
1/2      1-1005, 1025-4094

Port      Vlans allowed and active in management domain
-----
1/1      1-5,10,20,50,152,200,300,400,500,521-524,570,850,917,999
1/2      1-5,10,20,50,152,200,300,400,500,521-524,570,850,917,999
```

```

Port      Vlans in spanning tree forwarding state and not pruned
-----
1/1      1-5,10,20,50,152,200,300,400,500,521-524,570,850,917,999
1/2      1-5,10,20,50,152,200,300,400,500,521-524,570,850,917,999
Switch_A> (enable)

Switch_B> (enable) show trunk
Port      Mode          Encapsulation  Status      Native vlan
-----
3/1      auto          isl            trunking    1
3/2      auto          isl            trunking    1

Port      Vlans allowed on trunk
-----
3/1      1-1005, 1025-4094
3/2      1-1005, 1025-4094

Port      Vlans allowed and active in management domain
-----
3/1      1-5,10,20,50,152,200,300,400,500,521-524,570,850,917,999
3/2      1-5,10,20,50,152,200,300,400,500,521-524,570,850,917,999
Port      Vlans in spanning tree forwarding state and not pruned
-----
3/1      1-5,10,20,50,152,200,300,400,500,521-524,570,801,850,917,999
3/2      1-5,10,20,50,152,200,300,400,500,521-524,570,801,850,917,999
Switch_B> (enable)

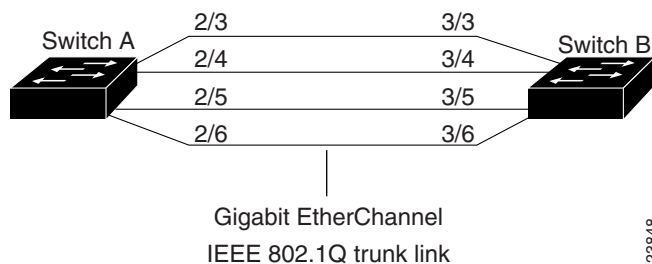
```

802.1Q Trunk Over EtherChannel Link Example

This example shows how to configure an 802.1Q trunk over an EtherChannel link between two switches.

Figure 5-2 shows two switches connected through four 1000BASE-SX Gigabit Ethernet ports.

Figure 5-2 802.1Q Trunk Over EtherChannel Link



This example shows how to configure the switches to form a four-port EtherChannel bundle and then configure the EtherChannel bundle as an 802.1Q trunk link.

- Step 1** Make sure that all ports on both Switch A and Switch B are assigned to the same VLAN by entering the **set vlan** command. This VLAN is used as the 802.1Q native VLAN for the trunk. In this example, all ports are configured as members of VLAN 1.

```
Switch_A> (enable) set vlan 1 2/3-6
VLAN Mod/Ports
-----
1     2/1-6
```

```
Switch_A> (enable)
```

```
Switch_B> (enable) set vlan 1 3/3-6
VLAN Mod/Ports
-----
1     3/1-6
```

```
Switch_B> (enable)
```

- Step 2** Confirm the channeling and trunking status of the switches by entering the **show port channel** and **show trunk** commands.

```
Switch_A> (enable) show port channel
No ports channelling
Switch_A> (enable) show trunk
No ports trunking.
Switch_A> (enable)
```

```
Switch_B> (enable) show port channel
No ports channelling
Switch_B> (enable) show trunk
No ports trunking.
Switch_B> (enable)
```

- Step 3** Configure the ports on Switch A to negotiate an EtherChannel bundle with the neighboring switch by entering the **set port channel** command. This example assumes that the neighboring ports on Switch B are in EtherChannel **auto** mode. The system logging messages provide information about the formation of the EtherChannel bundle.

```
Switch_A> (enable) set port channel 2/3-6 desirable
Port(s) 2/3-6 channel mode set to desirable.
Switch_A> (enable) %PAGP-5-PORTFROMSTP:Port 2/3 left bridge port 2/3
%PAGP-5-PORTFROMSTP:Port 2/4 left bridge port 2/4
%PAGP-5-PORTFROMSTP:Port 2/5 left bridge port 2/5
%PAGP-5-PORTFROMSTP:Port 2/6 left bridge port 2/6
%PAGP-5-PORTFROMSTP:Port 2/4 left bridge port 2/4
%PAGP-5-PORTFROMSTP:Port 2/5 left bridge port 2/5
%PAGP-5-PORTFROMSTP:Port 2/6 left bridge port 2/6
%PAGP-5-PORTFROMSTP:Port 2/3 left bridge port 2/3
%PAGP-5-PORTTOSTP:Port 2/3 joined bridge port 2/3-6
%PAGP-5-PORTTOSTP:Port 2/4 joined bridge port 2/3-6
%PAGP-5-PORTTOSTP:Port 2/5 joined bridge port 2/3-6
%PAGP-5-PORTTOSTP:Port 2/6 joined bridge port 2/3-6
```

```
Switch_B> (enable) %PAGP-5-PORTFROMSTP:Port 3/3 left bridge port 3/3
%PAGP-5-PORTFROMSTP:Port 3/4 left bridge port 3/4
%PAGP-5-PORTFROMSTP:Port 3/5 left bridge port 3/5
%PAGP-5-PORTFROMSTP:Port 3/6 left bridge port 3/6
%PAGP-5-PORTFROMSTP:Port 3/4 left bridge port 3/4
%PAGP-5-PORTFROMSTP:Port 3/5 left bridge port 3/5
%PAGP-5-PORTFROMSTP:Port 3/6 left bridge port 3/6
```

```
%PAGP-5-PORTFROMSTP:Port 3/3 left bridge port 3/3
%PAGP-5-PORTTOSTP:Port 3/3 joined bridge port 3/3-6
%PAGP-5-PORTTOSTP:Port 3/4 joined bridge port 3/3-6
%PAGP-5-PORTTOSTP:Port 3/5 joined bridge port 3/3-6
%PAGP-5-PORTTOSTP:Port 3/6 joined bridge port 3/3-6
```

Step 4 After the EtherChannel bundle is negotiated, verify the configuration by entering the **show port channel** command.

```
Switch_A> (enable) show port channel
Port Status      Channel  Channel  Neighbor  Neighbor
      mode      status   device   device
-----
2/3  connected  desirable channel   WS-C4003  JAB023806 (Sw 2/3
2/4  connected  desirable channel   WS-C4003  JAB023806 (Sw 2/4
2/5  connected  desirable channel   WS-C4003  JAB023806 (Sw 2/5
2/6  connected  desirable channel   WS-C4003  JAB023806 (Sw 2/6
-----
```

```
Switch_A> (enable)
```

```
Switch_B> (enable) show port channel
Port Status      Channel  Channel  Neighbor  Neighbor
      mode      status   device   device
-----
3/3  connected  auto     channel   WS-C4003  JAB023806 (Sw 2/3
3/4  connected  auto     channel   WS-C4003  JAB023806 (Sw 2/4
3/5  connected  auto     channel   WS-C4003  JAB023806 (Sw 2/5
3/6  connected  auto     channel   WS-C4003  JAB023806 (Sw 2/6
-----
```

```
Switch_B> (enable)
```

Step 5 Configure one of the ports in the EtherChannel bundle to negotiate an 802.1Q trunk by entering the **set trunk** command. The configuration is applied to all of the ports in the bundle. This example assumes that the neighboring ports on Switch B are configured to use **dot1q** or **negotiate** encapsulation and are in **auto** trunk mode. The system logging messages provide information about the formation of the 802.1Q trunk.

```
Switch_A> (enable) set trunk 2/3 desirable dot1q
Port(s) 2/3-6 trunk mode set to desirable.
Port(s) 2/3-6 trunk type set to dot1q.
Switch_A> (enable) %DTP-5-TRUNKPORTON:Port 2/3 has become dot1q trunk
%DTP-5-TRUNKPORTON:Port 2/4 has become dot1q trunk
%PAGP-5-PORTFROMSTP:Port 2/3 left bridge port 2/3-6
%DTP-5-TRUNKPORTON:Port 2/5 has become dot1q trunk
%PAGP-5-PORTFROMSTP:Port 2/4 left bridge port 2/3-6
%PAGP-5-PORTFROMSTP:Port 2/5 left bridge port 2/3-6
%DTP-5-TRUNKPORTON:Port 2/6 has become dot1q trunk
%PAGP-5-PORTFROMSTP:Port 2/6 left bridge port 2/3-6
%PAGP-5-PORTFROMSTP:Port 2/3 left bridge port 2/3
%PAGP-5-PORTTOSTP:Port 2/3 joined bridge port 2/3-6
%PAGP-5-PORTTOSTP:Port 2/4 joined bridge port 2/3-6
%PAGP-5-PORTTOSTP:Port 2/5 joined bridge port 2/3-6
%PAGP-5-PORTTOSTP:Port 2/6 joined bridge port 2/3-6
Switch_B> (enable) %DTP-5-TRUNKPORTON:Port 3/3 has become dot1q trunk
%DTP-5-TRUNKPORTON:Port 3/4 has become dot1q trunk
%PAGP-5-PORTFROMSTP:Port 3/3 left bridge port 3/3-6
%PAGP-5-PORTFROMSTP:Port 3/4 left bridge port 3/3-6
%PAGP-5-PORTFROMSTP:Port 3/5 left bridge port 3/3-6
%PAGP-5-PORTFROMSTP:Port 3/6 left bridge port 3/3-6
%DTP-5-TRUNKPORTON:Port 3/5 has become dot1q trunk
%DTP-5-TRUNKPORTON:Port 3/6 has become dot1q trunk
%PAGP-5-PORTFROMSTP:Port 3/5 left bridge port 3/3-6
%PAGP-5-PORTFROMSTP:Port 3/6 left bridge port 3/3-6
```

```
%PAGP-5-PORTTOSTP:Port 3/3 joined bridge port 3/3-6
%PAGP-5-PORTTOSTP:Port 3/4 joined bridge port 3/3-6
%PAGP-5-PORTTOSTP:Port 3/5 joined bridge port 3/3-6
%PAGP-5-PORTTOSTP:Port 3/6 joined bridge port 3/3-6
```

Step 6 After the 802.1Q trunk link is negotiated, verify the configuration by entering the **show trunk** command.

```
Switch_A> (enable) show trunk
Port      Mode           Encapsulation  Status        Native vlan
-----
2/3       desirable     dot1q          trunking      1
2/4       desirable     dot1q          trunking      1
2/5       desirable     dot1q          trunking      1
2/6       desirable     dot1q          trunking      1

Port      Vlans allowed on trunk
-----
2/3       1-1005, 1025-4094
2/4       1-1005, 1025-4094
2/5       1-1005, 1025-4094
2/6       1-1005, 1025-4094

Port      Vlans allowed and active in management domain
-----
2/3       1-5,10,20,50,152,200,300,400,500,521-524,570,850,917,999
2/4       1-5,10,20,50,152,200,300,400,500,521-524,570,850,917,999
2/5       1-5,10,20,50,152,200,300,400,500,521-524,570,850,917,999
2/6       1-5,10,20,50,152,200,300,400,500,521-524,570,850,917,999

Port      Vlans in spanning tree forwarding state and not pruned
-----
2/3
2/4
2/5
2/6
Switch_A> (enable)

Switch_B> (enable) show trunk
Port      Mode           Encapsulation  Status        Native vlan
-----
3/3       auto           dot1q          trunking      1
3/4       auto           dot1q          trunking      1
3/5       auto           dot1q          trunking      1
3/6       auto           dot1q          trunking      1

Port      Vlans allowed on trunk
-----
3/3       1-1005, 1025-4094
3/4       1-1005, 1025-4094
3/5       1-1005, 1025-4094
3/6       1-1005, 1025-4094

Port      Vlans allowed and active in management domain
-----
3/3       1-5,10,20,50,152,200,300,400,500,521-524,570,850,917,999
3/4       1-5,10,20,50,152,200,300,400,500,521-524,570,850,917,999
3/5       1-5,10,20,50,152,200,300,400,500,521-524,570,850,917,999
3/6       1-5,10,20,50,152,200,300,400,500,521-524,570,850,917,999
```

```

Port      Vlans in spanning tree forwarding state and not pruned
-----
3/3      1-5,10,20,50,152,200,300,400,500,521-524,570,850,917,999
3/4      1-5,10,20,50,152,200,300,400,500,521-524,570,850,917,999
3/5      1-5,10,20,50,152,200,300,400,500,521-524,570,850,917,999
3/6      1-5,10,20,50,152,200,300,400,500,521-524,570,850,917,999
Switch_B> (enable)

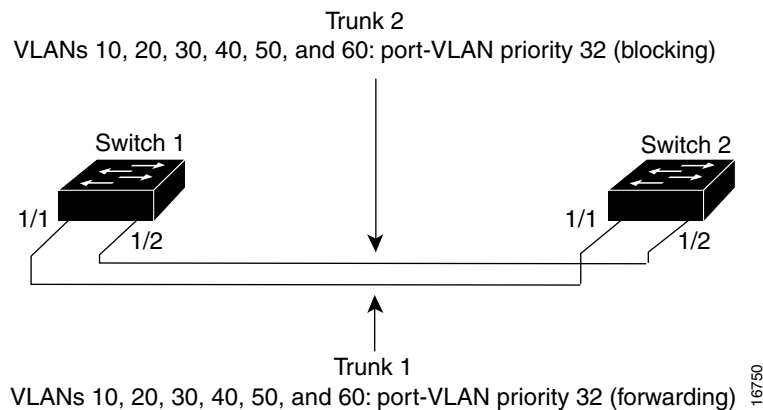
```

Load-Sharing VLAN Traffic Over Parallel Trunks Example

Using spanning tree port-VLAN priorities, you can load-share VLAN traffic over parallel trunk ports so that traffic from some VLANs travels over one trunk, while traffic from other VLANs travels over the other trunk. This configuration allows traffic to be carried over both trunks simultaneously (instead of keeping one trunk in blocking mode), which reduces the total traffic that is carried over each trunk while still maintaining a fault-tolerant configuration.

Figure 5-3 shows a parallel trunk configuration between two switches using the Fast Ethernet uplink ports on the supervisor engine.

Figure 5-3 Parallel Trunk Configuration Before Configuring VLAN-Traffic Load Sharing



By default, the port-VLAN priority for both trunks is equal (a value of 32). STP blocks port 1/2 (Trunk 2) for each VLAN on Switch 1 to prevent forwarding loops. Trunk 2 is not used to forward traffic unless Trunk 1 fails.

To configure the switches so that traffic from multiple VLANs is load balanced over the parallel trunks, perform these steps:

- Step 1** Configure a VTP domain on both Switch 1 and Switch 2 by entering the `set vtp` command so that the VLAN information configured on Switch 1 is learned by Switch 2. Make sure that Switch 1 is a VTP server. You can configure Switch 2 as a VTP client or as a VTP server.

```

Switch_1> (enable) set vtp domain BigCorp mode server
VTP domain BigCorp modified
Switch_1> (enable)

```

```

Switch_2> (enable) set vtp domain BigCorp mode server
VTP domain BigCorp modified
Switch_2> (enable)

```

- Step 2** Create the VLANs on Switch 1 by entering the **set vlan** command. In this example, you see VLANs 10, 20, 30, 40, 50, and 60.

```
Switch_1> (enable) set vlan 10
Vlan 10 configuration successful
Switch_1> (enable) set vlan 20
Vlan 20 configuration successful
Switch_1> (enable) set vlan 30
Vlan 30 configuration successful
Switch_1> (enable) set vlan 40
Vlan 40 configuration successful
Switch_1> (enable) set vlan 50
Vlan 50 configuration successful
Switch_1> (enable) set vlan 60
Vlan 60 configuration successful
Switch_1> (enable)
```

- Step 3** Verify the VTP and VLAN configuration on Switch 1 by entering the **show vtp domain** and **show vlan** commands.

```
Switch_1> (enable) show vtp domain
Domain Name                Domain Index VTP Version Local Mode Password
-----
BigCorp                    1            2            server      -

Vlan-count Max-vlan-storage Config Revision Notifications
-----
11          1023             13           disabled

Last Updater    V2 Mode  Pruning  PruneEligible on Vlans
-----
172.20.52.10   disabled enabled   2-1000
Switch_1> (enable) show vlan
VLAN Name                Status      Mod/Ports, Vlans
-----
1    default                active     1/1-2
                                   2/1-12
                                   5/1-2
10   VLAN0010                active
20   VLAN0020                active
30   VLAN0030                active
40   VLAN0040                active
50   VLAN0050                active
60   VLAN0060                active
1002 fddi-default            active
1003 token-ring-default    active
1004 fddinet-default        active
1005 trnet-default          active
.
.
.

Switch_1> (enable)
```

- Step 4** Configure the supervisor engine uplinks on Switch 1 as ISL trunk ports by entering the **set trunk** command. Specifying the **desirable** mode on the Switch 1 ports causes the ports on Switch 2 to negotiate to become trunk links (assuming that the Switch 2 uplinks are in the default **auto** mode).

```
Switch_1> (enable) set trunk 1/1 desirable
Port(s) 1/1 trunk mode set to desirable.
Switch_1> (enable) 04/21/1998,03:05:05:DISL-5:Port 1/1 has become isl trunk
```

```
Switch_1> (enable) set trunk 1/2 desirable
Port(s) 1/2 trunk mode set to desirable.
Switch_1> (enable) 04/21/1998,03:05:13:DISL-5:Port 1/2 has become isl trunk
```

Step 5 Verify that the trunk links are up by entering the **show trunk** command.

```
Switch_1> (enable) show trunk 1
Port      Mode           Encapsulation  Status      Native vlan
-----
1/1      desirable     isl            trunking    1
1/2      desirable     isl            trunking    1

Port      Vlans allowed on trunk
-----
1/1      1-1005, 1025-4094
1/2      1-1005, 1025-4094

Port      Vlans allowed and active in management domain
-----
1/1      1,10,20,30,40,50,60
1/2      1,10,20,30,40,50,60

Port      Vlans in spanning tree forwarding state and not pruned
-----
1/1
1/2
Switch_1> (enable)
```

Step 6 Note that when the trunk links come up, VTP passes the VTP and VLAN configuration to Switch 2. Verify that Switch 2 has learned the VLAN configuration by entering the **show vlan** command on Switch 2.

```
Switch_2> (enable) show vlan
VLAN Name                Status      Mod/Ports, Vlans
-----
1    default                active
10   VLAN0010               active
20   VLAN0020               active
30   VLAN0030               active
40   VLAN0040               active
50   VLAN0050               active
60   VLAN0060               active
1002 fddi-default           active
1003 token-ring-default    active
1004 fddinet-default       active
1005 trnet-default        active
.
.
.
Switch_2> (enable)
```

Step 7 Note that spanning tree takes one to two minutes to converge. After the network stabilizes, check the spanning tree state of each trunk port on Switch 1 by entering the **show spantree** command. Trunk 1 is forwarding for all VLANs. Trunk 2 is blocking for all VLANs. On Switch 2, both trunks are forwarding for all VLANs, but no traffic passes over Trunk 2 because port 1/2 on Switch 1 is blocking.

```

Switch_1> (enable) show spantree 1/1
Port      Vlan  Port-State  Cost  Priority  Fast-Start  Group-method
-----
1/1       1    forwarding  19    32       disabled
1/1       10   forwarding  19    32       disabled
1/1       20   forwarding  19    32       disabled
1/1       30   forwarding  19    32       disabled
1/1       40   forwarding  19    32       disabled
1/1       50   forwarding  19    32       disabled
1/1       60   forwarding  19    32       disabled
1/1       1003 not-connected 19    32       disabled
1/1       1005 not-connected 19    4        disabled
Switch_1> (enable) show spantree 1/2
Port      Vlan  Port-State  Cost  Priority  Fast-Start  Group-method
-----
1/2       1    blocking    19    32       disabled
1/2       10   blocking    19    32       disabled
1/2       20   blocking    19    32       disabled
1/2       30   blocking    19    32       disabled
1/2       40   blocking    19    32       disabled
1/2       50   blocking    19    32       disabled
1/2       60   blocking    19    32       disabled
1/2       1003 not-connected 19    32       disabled
1/2       1005 not-connected 19    4        disabled
Switch_1> (enable)

```

- Step 8** Divide the configured VLANs into two groups. You might want traffic from half of the VLANs to go over one trunk link and half over the other, or if one VLAN has heavier traffic than the others, you can forward traffic from that VLAN over one trunk and traffic from the other VLANs over the other trunk link.



Note In the following steps, VLANs 10, 20, and 30 (Group 1) are forwarded over Trunk 1, and VLANs 40, 50, and 60 (Group 2) are forwarded over Trunk 2.

- Step 9** On Switch 1, change the port-VLAN priority for the Group 1 VLANs on Trunk 1 (port 1/1) to an integer value lower than the default of 32 by entering the **set spantree portvlanpri** command.

```

Switch_1> (enable) set spantree portvlanpri 1/1 1 10
Port 1/1 vlans 1-9,11-1004 using portpri 32.
Port 1/1 vlans 10 using portpri 1.
Port 1/1 vlans 1005 using portpri 4.
Switch_1> (enable) set spantree portvlanpri 1/1 1 20
Port 1/1 vlans 1-9,11-19,21-1004 using portpri 32.
Port 1/1 vlans 10,20 using portpri 1.
Port 1/1 vlans 1005 using portpri 4.
Switch_1> (enable) set spantree portvlanpri 1/1 1 30
Port 1/1 vlans 1-9,11-19,21-29,31-1004 using portpri 32.
Port 1/1 vlans 10,20,30 using portpri 1.
Port 1/1 vlans 1005 using portpri 4.
Switch_1> (enable)

```

- Step 10** On Switch 1, change the port-VLAN priority for the Group 2 VLANs on Trunk 2 (port 1/2) to an integer value lower than the default of 32 by entering the **set spantree portvlanpri** command.

```

Switch_1> (enable) set spantree portvlanpri 1/2 1 40
Port 1/2 vlans 1-39,41-1004 using portpri 32.
Port 1/2 vlans 40 using portpri 1.
Port 1/2 vlans 1005 using portpri 4.

```

```
Switch_1> (enable) set spantree portvlanpri 1/2 1 50
Port 1/2 vlans 1-39,41-49,51-1004 using portpri 32.
Port 1/2 vlans 40,50 using portpri 1.
Port 1/2 vlans 1005 using portpri 4.
Switch_1> (enable) set spantree portvlanpri 1/2 1 60
Port 1/2 vlans 1-39,41-49,51-59,61-1004 using portpri 32.
Port 1/2 vlans 40,50,60 using portpri 1.
Port 1/2 vlans 1005 using portpri 4.
Switch_1> (enable)
```

- Step 11** On Switch 2, change the port-VLAN priority for the Group 1 VLANs on Trunk 1 (port 1/1) to the same value that you configured for those VLANs on Switch 1 by entering the **set spantree portvlanpri** command.

**Caution**

The port-VLAN priority for each VLAN must be equal on both ends of the link.

```
Switch_2> (enable) set spantree portvlanpri 1/1 1 10
Port 1/1 vlans 1-9,11-1004 using portpri 32.
Port 1/1 vlans 10 using portpri 1.
Port 1/1 vlans 1005 using portpri 4.
Switch_2> (enable) set spantree portvlanpri 1/1 1 20
Port 1/1 vlans 1-9,11-19,21-1004 using portpri 32.
Port 1/1 vlans 10,20 using portpri 1.
Port 1/1 vlans 1005 using portpri 4.
Switch_2> (enable) set spantree portvlanpri 1/1 1 30
Port 1/1 vlans 1-9,11-19,21-29,31-1004 using portpri 32.
Port 1/1 vlans 10,20,30 using portpri 1.
Port 1/1 vlans 1005 using portpri 4.
Switch_2> (enable)
```

- Step 12** On Switch 2, change the port-VLAN priority for the Group 2 VLANs on Trunk 2 (port 1/2) to the same value that you configured for those VLANs on Switch 1 by entering the **set spantree portvlanpri** command.

```
Switch_2> (enable) set spantree portvlanpri 1/2 1 40
Port 1/2 vlans 1-39,41-1004 using portpri 32.
Port 1/2 vlans 40 using portpri 1.
Port 1/2 vlans 1005 using portpri 4.
Switch_2> (enable) set spantree portvlanpri 1/2 1 50
Port 1/2 vlans 1-39,41-49,51-1004 using portpri 32.
Port 1/2 vlans 40,50 using portpri 1.
Port 1/2 vlans 1005 using portpri 4.
Switch_2> (enable) set spantree portvlanpri 1/2 1 60
Port 1/2 vlans 1-39,41-49,51-59,61-1004 using portpri 32.
Port 1/2 vlans 40,50,60 using portpri 1.
Port 1/2 vlans 1005 using portpri 4.
Switch_2> (enable)
```

**Note**

When you have configured the port-VLAN priorities on both ends of the link, the spanning tree converges to use the new configuration.

- Step 13** Check the spanning tree port states on Switch 1 by entering the **show spantree** command. The Group 1 VLANs should forward on Trunk 1 and block on Trunk 2. The Group 2 VLANs should block on Trunk 1 and forward on Trunk 2.

If Trunk 1 fails in the network shown in [Figure 5-4](#), STP reconverges to use Trunk 2 to forward traffic from all the VLANs, as shown in this example:

```
Switch_1> (enable) 04/21/1998,03:15:40:DISL-5:Port 1/1 has become non-trunk
```

```
Switch_1> (enable) show spantree 1/1
Port      Vlan  Port-State  Cost  Priority  Fast-Start  Group-method
-----
1/1       1     not-connected  19    32      disabled
Switch_1> (enable) show spantree 1/2
Port      Vlan  Port-State  Cost  Priority  Fast-Start  Group-method
-----
1/2       1     learning    19    32      disabled
1/2       10    learning    19    32      disabled
1/2       20    learning    19    32      disabled
1/2       30    learning    19    32      disabled
1/2       40    forwarding   19    1       disabled
1/2       50    forwarding   19    1       disabled
1/2       60    forwarding   19    1       disabled
1/2      1003  not-connected  19    32      disabled
1/2      1005  not-connected  19    4       disabled
Switch_1> (enable) show spantree 1/2
Port      Vlan  Port-State  Cost  Priority  Fast-Start  Group-method
-----
1/2       1     forwarding   19    32      disabled
1/2       10    forwarding   19    32      disabled
1/2       20    forwarding   19    32      disabled
1/2       30    forwarding   19    32      disabled
1/2       40    forwarding   19    1       disabled
1/2       50    forwarding   19    1       disabled
1/2       60    forwarding   19    1       disabled
1/2      1003  not-connected  19    32      disabled
1/2      1005  not-connected  19    4       disabled
Switch_1> (enable)
```