



Configuring Interfaces and Circuits

This chapter describes how to configure the CSS interfaces and circuits, and bridge interfaces to VLANs. Information in this chapter applies to all CSS models except where noted.

This chapter contains the following sections:

- Interface and Circuit Overview
- Configuring Interfaces
- Configuring Circuits
- Configuring a Circuit IP Interface
- Configuring RIP for an IP Interface

Interface and Circuit Overview

The CSS provides ethernet interfaces (ports) that enable you to connect servers, PCs, routers, and other devices to the CSS.

Using the **bridge** command, you bridge (assign) the interfaces to a specific Virtual LAN (VLAN). Each VLAN circuit requires an IP address. Assigning an IP address to each VLAN circuit allows the CSS to route Ethernet interfaces from VLAN to VLAN.

Using the **trunk** command, you can assign multiple VLANs to a CSS interface port (Gigabit Ethernet ports only). A trunk is a point-to-point link carrying the traffic of several VLANs (as illustrated in Figure 4-2). The advantage of a trunk is to save ports when creating a link between two devices implementing VLANs, typically two switches. A trunk bundles virtual links over one physical link. The unique physical link between the two CSSs is able to carry traffic for the specified VLANs.

**Note**

The **trunk** and **vlan** commands (and the associated software functionality) comply with the IEEE 802.1Q Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks.

The CSS forwards VLAN circuit traffic to the IP interface. The IP interface passes the traffic to the IP forwarding function where the CSS compares the destination of each packet to information contained in the routing table. Once the CSS resolves the packet addresses, it forwards the packet to the appropriate VLAN and destination port.

When trunking is enabled, the CSS automatically inserts a tag in every frame transmitted over the trunk link to identify the originating VLAN. When the VLAN-aware device receives the frame the device reviews the VLAN-tagged packet to identify the transmitting VLAN. If the VLAN is recognized, the frame is routed to the proper port and VLAN destination. If the frame is from a VLAN that is not assigned to the trunk port, the packet is ignored. By default, the CSS discards untagged packets.

**Note**

A VLAN-tagged frame is a frame that contains a 4-byte Tag Header immediately following the Source MAC address field in the frame. Each VLAN-tagged frame carries an explicit identification of the VLAN to which it belongs.

Note that on an 802.1Q trunk, you may configure one VLAN (using the optional **default-vlan** command) to:

- Accept packets that arrive untagged on the interface
- Transmit untagged packets

By this method, the CSS can determine which VLAN transmitted an untagged frame. This capability allows VLAN-aware devices and VLAN-unaware devices to transmit and receive information on the same cable.

Figure 4-1 illustrates the interfaces, circuits, and VLANs in a CSS and Figure 4-2 illustrates trunking between VLANs.

Figure 4-1 Content Services Switch Interfaces and Circuits

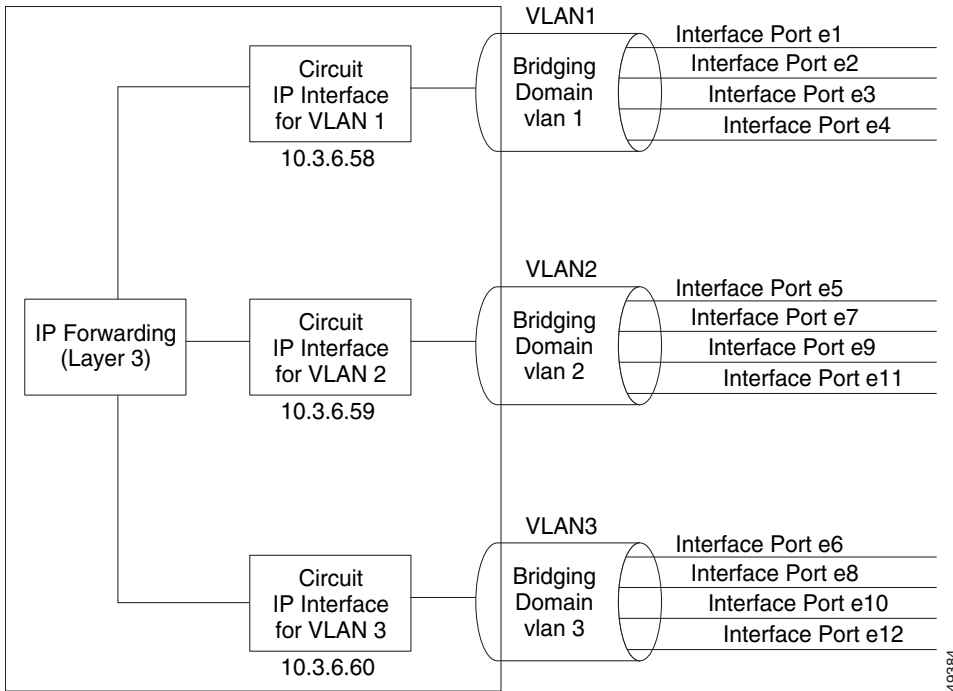
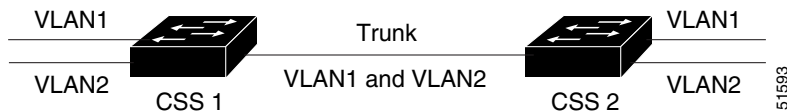


Figure 4-2 Interface Trunking Between VLANs



Interface and Circuit Configuration Quick Start

Table 4-1 provides a quick overview of the steps required to configure interfaces and circuits. Each step includes the CLI command required to complete the task. For a complete description of each feature and all the options associated with the CLI command, refer to the sections following Table 4-1.

Table 4-1 Interface and Circuit Configuration Quick Start

Task and Command Example

1. Log into the CSS.

2. Enter into config mode by typing **config**.

```
# config
(config)#
```

3. Enter into the interface mode for the interface you wish to configure.

This set of interface commands applies to the CSS 11050 and CSS 11150.

```
CSS11150(config)# interface e1
CSS11150(config-if[e1])#
```

This set of interface commands applies to the CSS 11800.

```
CSS11800(config)# interface 2/1
CSS11800(config-if[2/1])#
```

4. Configure the interface duplex and speed if required (default is auto-negotiate).

```
CSS11800(config-if[2/1])# phy 1Gbits-FD-no-pause
```

5. Bridge the interface to a VLAN. All interfaces are assigned to VLAN1 by default.

```
CSS11800(config-if[2/1])# bridge vlan 2
```

6. Enable trunking for a CSS Gigabit Interface (optional).

```
CSS11800(config-if[2/1])# trunk
CSS11800(config-if[2/1])# vlan 2
Create VLAN<2>, [y/n]:y
CSS11800(config-if-vlan[2/1-2])# vlan 3
Create VLAN<3>, [y/n]:y
CSS11800(config-if-vlan[2/1-3])#
```

Table 4-1 Interface and Circuit Configuration Quick Start (continued)

Task and Command Example

7. Display all circuit information for circuits that are currently active (optional).

```
CSS11800(config-if[2/1])# show circuit all
```

8. Display the interface configuration (optional).

```
CSS11800(config-if[2/1])# show interface
CSS11800(config-if[2/1])# exit
```

9. Configure circuits as required. Assign an IP address and subnet mask to each circuit.

```
CSS11800(config)# circuit VLAN1
CSS11800(config-circuit[VLAN1])# ip address 10.3.6.58/24
CSS11800(config)# circuit VLAN3
CSS11800(config-circuit[VLAN3])# ip address 10.3.6.60/24
CSS11800(config-circuit-ip[VLAN3-10.3.6.60])# exit
```

10. Display the circuit configuration (optional).

```
CSS11800(config-circuit[VLAN1])# show circuit all
```

11. Save your configuration changes (recommended). If you do not save the running-config, all configuration changes are lost upon reboot.

```
CSS11800(config)# copy running-config startup-config
```

Configuring Interfaces

Interfaces are ports that enable you to connect devices to the CSS and connect the CSS to the Internet. The commands to configure interfaces on the CSS 11050 and CSS 11150 differ slightly from the commands to configure interfaces on the CSS 11800 because the CSS 11800 requires a slot/port designation. The CSS 11050 and CSS 11150 do not use the slot/port designation.

Configuring an interface requires you to:

- Configure an interface and enter into the interface mode
- Add a description of the interface (optional)
- Configure an interface duplex and speed
- Set the maximum idle time for the interface
- Bridge the interface to a VLAN, or specify VLAN trunking to the interface
- Configure the low water mark of flow control blocks (optional)
- Smoothing bursty network traffic on the CSS 11800 Gigabit Ethernet Module (optional)

**Note**

Only use the **gem-traffic-bursty** command when it is necessary to smooth bursty traffic (when traffic bursts over a rate of 16 Gbps for short intervals) on the CSS 11800 Gigabit Ethernet Modules for applications that are sensitive to packet loss.

Configuring an Interface

To configure an interface, enter the **interface** command and a valid interface name. To use the **interface** command for a:

- CSS 11050 or CSS 11150, enter the interface name in *interface port* format (for example, e2)
- CSS 11800, enter the interface format in *slot/port* format (for example, 3/1)

For example, to configure interface e1 on a CSS 11050 or CSS 11150, access interface mode for e1 by entering:

```
(config)# interface e1
```

The CSS changes from config mode to the specific interface mode.

```
(config-if[e1])#
```

Entering a Description for the Interface

To help you to identify the interface, use the **description** command to describe the interface. Enter a quoted text string from 1 to 255 characters in length including spaces.

For example:

```
(config-if[e1])# description "Connects to server17"
```

To display an interface description, use the **show running-config interface** command. For example:

```
(config-if[e1])# show running-config interface e11  
  
!***** INTERFACE *****  
interface e11  
    description "Connects to server17"  
    bridge vlan 2
```

To remove an interface description, enter:

```
(config-if[e1])# no description
```

Configuring Interface Duplex and Speed

Use the **phy** command to configure the speed or flow control (pause) method and duplex for a CSS Fast Ethernet or Gigabit Ethernet interface (port), respectively. By default, the port is configured to auto-negotiate. Auto-negotiate enables the port to detect the speed or pause method, and duplex of incoming signals and synchronize with them automatically.

For Fast Ethernet modules (FEM), when older equipment cannot transmit the duplex and speed with its signals, you can configure the speed and duplex on the port to match the transmitting equipment.

For Gigabit Ethernet modules (GEM), if the link does not come up and you need to force the module and its link partner into a specific mode, you can set the duplex and flow control (pause) method. The pause methods of communications between modules determines how the module sends pause frames. The module sends pause frames when it becomes overwhelmed with data. The CSS module and its link partner must be configured with the same pause method.

Use the **phy** command to manually configure the interface (port) duplex and speed to one of the following settings:

- **phy 10Mbits-FD** - Set the FEM port to 10 Mbits per second and full-duplex.
- **phy 10Mbits-HD** - Set the FEM port to 10 Mbits per second and half-duplex.
- **phy 100Mbits-FD** - Set the FEM port to 100 Mbits per second and full-duplex.
- **phy 100Mbits-HD** - Set the FEM port to 100 Mbits per second and half-duplex.
- **phy auto-negotiate** - Resets the FEM or GEM port to automatically negotiate speed or pause method, respectively, and duplex (default).
- **phy 1Gbits-FD-async** - Set the GEM port to full duplex mode with asymmetric pause toward the link partner.
- **phy 1Gbits-FD-no pause** - Set the GEM port to full duplex mode with no pause.
- **phy 1Gbits-FD-sym** - Set the GEM port to full duplex mode with symmetric pause.
- **phy 1Gbits-FD-sym-async** - Set the GEM port to full duplex mode with asymmetric and symmetric pause toward the local device.

For example, to set interface e1 to 100 Mbps and half-duplex, enter:

```
(config-if[e1])# phy 100Mbits-HD
```

Setting the Interface Maximum Idle Time

Use the **max-idle** command as a troubleshooting tool to verify an interface's ability to receive traffic. If the interface does not receive traffic within the configured idle time, the CSS reinitializes it automatically.

Set the idle time to a value greater than the interval over which the interface is not receiving traffic. For example, if the interface receives traffic every 90 seconds, set the idle time to a value greater than 90 seconds. If you set the idle time to less than 90 seconds, the CSS would continuously reinitialize the interface before the interface was able to receive traffic.

Enter the idle time from 15 to 65535 seconds. The default is 0, which disables the idle timer. For example, to set the maximum idle time for interface e7 to 180 seconds, enter:

```
(config-if[e7])# max-idle 180
```

To reset the idle time for an interface to its default value of 0, enter:

```
(config-if[e7])# no max-idle
```

Showing Interface Duplex and Speed

To show duplex and speed values for all interfaces, enter the **show phy** command. For example:

```
(config)# show phy
```

To show duplex and speed value for a specific interface, enter the **show phy** command and the interface name. For example:

```
(config)# show phy e3
```

Table 4-2 describes the fields in the **show phy** output.

Table 4-2 Field Descriptions for the **show phy** Command

Field	Description
Name	The name of the physical interface.
Configured Speed	The configured speed for the Ethernet interface (port) in the CSS. Auto indicates that the speed is automatically negotiated.

Table 4-2 Field Descriptions for the `show phy` Command (continued)

Field	Description
Configured Duplex	The configured duplex for the Ethernet interface (port) in the CSS. Auto indicates that the duplex is automatically negotiated.
Actual Speed	The actual speed for the Ethernet interface (port) in the CSS.
Actual Duplex	The configured duplex for the Ethernet interface (port) in the CSS.
Link	The link status, up or down.
Rev	Revision number of the chip.
Partner Auto	Whether or not auto-negotiation is available on the link partner.

Bridging an Interface to a VLAN

To bridge an interface to a VLAN, use the **bridge vlan** command to specify a virtual local area network (VLAN) and associate it with the specified interface port. Enter an integer from 1 to 4094 as the VLAN identifier. The default is 1. All interfaces are assigned to VLAN1 by default.



Note

The CSS 11050 and CSS 11150 both support a maximum of 16 VLANs. The CSS 11800 supports a maximum of 128 VLANs.

For example, to configure e1 to VLAN2, enter:

```
(config-if[e1])# bridge vlan 2
```



Note

When you enter the **bridge vlan** command, enter the word **vlan** in lowercase letters and include a space before the VLAN number (for example, **vlan 2**).

**Note**

The CSS Gigabit Ethernet interface supports trunking interfaces to multiple VLANs through the **trunk** command. The **trunk** command would be used for the ethernet interface instead of **bridge vlan** (and the other associated bridge CLI commands). Refer to “Specifying VLAN Trunking to an Interface” for details.

To restore the default VLAN1, enter:

```
(config-if[e7])# no bridge vlan
```

To display all interfaces and the VLANs to which they are configured, enter the **show circuit** command. Refer to “Showing Circuits” in this chapter for information about the **show circuits** command.

In the **show circuit** display, VLANs appear as VLAN (uppercase, with no space before the VLAN number).

Configuring Bridge Pathcost

To set the path cost for an interface, use the **bridge pathcost** command. The cost is the contribution of the interface to the vast path cost towards spanning-tree root. Enter an integer from 1 to 65535. The default is dynamically configured based on the interface speed.

For example, to set a path cost of 9 for e7, enter:

```
(config-if[e7])# bridge pathcost 9
```

To restore the default path cost, enter:

```
(config-if[e7])# no bridge pathcost
```

Configuring Bridge Priority

To set the bridge priority for a port, use the **bridge priority** command. Enter an integer from 0 to 255. The default is 128.

For example, to set a bridge priority of 100 for e7, enter:

```
(config-if[e7])# bridge priority 100
```

To restore the default priority of 128, enter:

```
(config-if[e7])# no bridge priority
```

Configuring Bridge State

To set the bridge state to enable or disable for an interface, use the **bridge state** command. An interface is set to bridge state **enable** by default.

For example, to enable the bridge state for e7, enter:

```
(config-if[e7])# bridge state enable
```

To disable the bridge state for e7, enter:

```
(config-if[e7])# bridge state disable
```

Specifying VLAN Trunking to an Interface

To activate VLAN trunking for a CSS Gigabit Interface, use the **trunk** command and specify all VLANs that are to include the specified port as part of the VLAN. The **trunk** command also converts the link into a trunk link. To specify the number of each VLAN that is to be associated with the Gigabit Interface, use the **vlan** command. Enter an integer from 1 to 4094 as the VLAN identifier.



Note

The CSS 11050 and CSS 11150 both support a maximum of 16 VLANs. The CSS 11800 supports a maximum of 128 VLANs.



Note

The CSS software has a dependency when using the **trunk** command. For trunking to be enabled, all VLAN bridging commands for any active VLAN must first be disabled for the Gigabit Interface by using the **no bridge vlan**, **no bridge priority**, **no bridge state**, and **no bridge pathcost** commands. If you do not disable VLAN bridging on an interface, the CSS software instructs you to do so.

**Note**

When you enter the **trunk** command, enter the word **vlan** in lowercase letters and include a space before the VLAN number (for example, **vlan 2**).

For example, to configure gigabit ethernet port 1 in slot 1 for use in VLAN2, VLAN3, and VLAN9, enter:

```
CSS11800(config-if[1/1])# trunk
CSS11800(config-if[1/1])# vlan 2
Create VLAN<2>, [y/n]:y
CSS11800(config-if-vlan[1/1-2])# vlan 3
Create VLAN<3>, [y/n]:y
CSS11800(config-if-vlan[1/1-3])# vlan 9
Create VLAN<9>, [y/n]:y
CSS11800(config-if-vlan[1/1-9])#
```

Every time you enter a **vlan** command for a new VLAN, the software automatically prompts you to create the specified VLAN (where **y** instructs the software to create the VLAN and **n** cancels the VLAN creation).

To disable trunking on the specified Gigabit Interface and associated VLANs, enter:

```
(config-trunkif[2/3])# no trunk
```

The **no trunk** command turns off all trunking, removes all specified **vlan** commands associated with the interface, and deletes this information from the running configuration. The interface is returned to VLAN1 by default.

To display all interfaces and the VLANs to which they are configured, enter the **show circuit** command. Refer to Table 4-9 for information about the **show circuits** command.

**Note**

In the **show circuit** display, VLANs appear as VLAN (uppercase, with no space before the VLAN number). For an interface that has trunking enabled, a “-*n*” (where *n* is the associated VLAN number) is appended to the prefix. In this example, 1/4-1 indicates slot 1, port 4, VLAN1.

Selecting a Default VLAN in a Trunk

To define a default VLAN that has the capability to accept packets that arrive untagged on the interface, include the **default-vlan** command as part of the trunk/VLAN definition. The command also specifies that the packets transmitted from this VLAN will be untagged. The default VLAN must be explicitly set if you want untagged packets to be processed by the CSS. Otherwise these packets will be discarded.

The **default-vlan** command can only be specified for a single VLAN. If you attempt to use this command for another VLAN, the software instructs you to disable the current default VLAN (using the **no default-vlan** command).

For example, enter:

```
CSS11800(config-if[1/1])# trunk
CSS11800(config-if[1/1])# vlan 2
Create VLAN<2>, [y/n]:y
CSS11800(config-if-vlan[1/1-2])# vlan 3
Create VLAN<3>, [y/n]:y
CSS11800(config-if-vlan[1/1-3])# default-vlan
```

To remove the default VLAN selection, enter:

```
CSS11800(config-if-vlan[1/1-3])# no default-vlan
```

Configuring Bridge Pathcost for a Trunked Interface/VLAN Pair

To set the spanning-tree port path cost for a given interface/VLAN pair, use the **bridge pathcost** command. The cost is the contribution of the given interface/VLAN pair to the vast path cost towards spanning-tree root. Enter an integer from 1 to 65535. The default is dynamically configured based on the link speed.

For example, to set a path cost of 2 for slot 1, port 1, VLAN3, enter:

```
CSS11800(config-if-vlan[1/1-3])# bridge pathcost 2
```

To restore the default path cost, enter:

```
CSS11800(config-if-vlan[1/1-3])# no bridge pathcost
```

Configuring Bridge Priority for a Trunked Interface/VLAN Pair

To set the spanning-tree bridge priority for a given interface/VLAN pair, use the **bridge priority** command. Enter an integer from 0 to 255. The default is 128.

For example, to set a bridge priority of 100 for slot 1, port 1, VLAN3, enter:

```
CSS11800(config-if-vlan[1/1-3])# bridge priority 100
```

To restore the default priority of 128, enter:

```
CSS11800(config-if-vlan[1/1-3])# no bridge priority
```

Configuring Bridge State for a Trunked Interface/VLAN Pair

To set the bridge state to enable or disable for a given interface/VLAN pair, use the **bridge state** command. An interface is set to bridge state **enable** by default.

For example, to enable the bridge state for slot 1, port 1, VLAN3, enter:

```
CSS11800(config-if-vlan[1/1-3])# bridge state enable
```

To disable the bridge state for slot 1, port 1, located in VLAN3, enter:

```
CSS11800(config-if-vlan[1/1-3])# bridge state disable
```

Configuring the Low-Water Mark of Flow Control Blocks on an Interface

Use the **fcblowwater** command to configure the low-water mark of flow control blocks (FCBs) on the interface. The low-water mark is the percentage of the total number of FCBs available. If the number of FCBs available on a port goes below the low-water mark, then aggressive flow recovery occurs.

The syntax for this interface mode command is:

```
fcblowwater percentage
```

Enter the *percentage* as the total number of available FCBs. Enter a number from 1 to 100. The default is 25%.

To reset the percentage of available FCBs to its default of 25, enter:

```
(config-if[e7])# no fcblowwater
```

Smoothing Bursty Network Traffic on the CSS 11800 Gigabit Ethernet Module

Use the **gem-traffic-bursty** global configuration mode command to smooth bursty traffic on Gigabit Ethernet Modules (GEMs) in the CSS 11800 for applications sensitive to packet loss. Traffic burstiness is the occurrence of extreme amounts of traffic for a short period of time (when traffic bursts over a rate of 16 Gbps for short intervals). During extremely heavy traffic loads, when a single GEM port has greater than one gigabit per second of incoming network traffic, substantial packet loss can occur. This condition can easily occur when a group of servers attached to multiple ports send traffic simultaneously to a single client uplink port.

If the traffic load at the client uplink port is at a rate close to a gigabit per second with occasional bursts of greater than one gigabit per second, you can use the **gem-traffic-bursty** command to reduce overall packet loss. This command can greatly reduce packet loss for applications sensitive to this condition, for example, video and audio streaming applications.

If the traffic load at the client uplink port remains at a constant rate greater than one gigabit per second, you may need to perform a network reconfiguration, for example, configure an additional client uplink port. You should not use the **gem-traffic-bursty** command to solve the problem.

**Note**

When you issue the **gem-traffic-bursty** command it applies to all GEM modules in the CSS 11800 chassis (installed prior to and after you issue the command).

To smooth bursty traffic on the GEMs in a CSS 11800, enter:

```
(config)# gem-traffic-bursty
```

To reset the default traffic handling behavior on the GEM, enter:

```
(config)# no gem-traffic-bursty
```

Showing Bridge Configurations

The CSS enables you to show bridging information. To display this information, use the **show bridge** command.

The syntax for this interface mode command is:

```
show bridge [forwarding|status] {vlan_number}
```

The options and variables are:

- **forwarding** - Displays the bridge forwarding table including the VLAN number, the MAC addresses, and port numbers.
- **status** - Displays the bridge spanning-tree status including the STP state, designated root, bridge ID, and root maximum age, hello time and forward delay, and port information including state, VLAN, root and port cost, and designated root and port number.
- *vlan_number* - Displays the forwarding table or spanning tree status for the specified VLAN number. To see a list of VLAN numbers, enter **show bridge [forwarding|status] ?**

Table 4-3 describes the fields in the **show bridge forwarding** output.

Table 4-3 Field Descriptions for the show bridge forwarding Command

Field	Description
VLAN	The bridge interface virtual LAN number
MAC Address	The MAC address for the entries
Port Number	The port number for the forwarding

To display bridge status information, use the **show bridge status** command.



Note

To display bridge forwarding or bridge status for a specific VLAN in the CSS, enter the **show bridge forwarding** or **show bridge status** command with the VLAN number. Typing the **show bridge** command with a VLAN number returns a list of available VLANs.

Table 4-4 describes the fields in the **show bridge status** output.

Table 4-4 Field Descriptions for the show bridge status Command

Field	Description
STP State	The state of the spanning-tree protocol, enabled or disabled.
Root Max Age	The timeout period in seconds of the host for timing out root information.
Root Hello Time	The interval in seconds that the root broadcasts its hello message to other switches.
Root Fwd Delay	The delay time in seconds that the root uses for forward delay.
Designated Root	The bridge ID for the designated root.
Bridge ID	The bridge ID of this bridge.
Port	The port ID.
State	<p>The state of the port. The possible states are:</p> <ul style="list-style-type: none"> • Block, the blocking state. A port enters the blocking state after switch initialization. The port does not participate in frame forwarding. • Listen, the listening state. This state is the first transitional state a port enters after the blocking state. The port enters this state when STP determines that the port should participate in frame forwarding. • Learn, the learning state. The port enters the learning state from the listening state. The port in the learning state prepares to participate in frame forwarding. • Forward, the forwarding state. The port enters the forwarding state from the learning state. A port in the forwarding state forwards frames. • Disabled, the disabled state. A port in the disabled state does not participate in frame forwarding or STP. A port in the disabled state is non operational.
Designated Bridge	The bridge ID for the designated bridge.
Designated Root	The bridge ID for the designated root.

Table 4-4 Field Descriptions for the `show bridge status` Command (continued)

Field	Description
Root Cost	The cost of the root.
Port Cost	The cost of the port.
Desg Port	Designated port.

Showing Trunking Configurations

The CSS enables you to show VLAN trunk status information for a Gigabit Ethernet port. To display this information, use the **show trunk** command.

Table 4-5 describes the fields in the **show trunk** output.

Table 4-5 Field Description for the `show trunk` Command

Field	Description
Port	The CSS port
VLAN	The VLAN on the port
Default	The configured default VLAN on the port (if there is not a configured default VLAN, “None” appears in this field)

Showing Interfaces

To display a list of valid interfaces for the CSS, use the **show interface** command. For example:

```
(config)# show interface
```

To only display information for a specific interface, enter the **show interface** command and the interface name.

```
(config)# show interface e7
```

Table 4-6 describes the fields in the **show interface** output.

Table 4-6 *Field Descriptions for the show interface Command*

Field	Description
Name	The name of the interface.
ifIndex	The ifIndex for the interface.
Type	The type of interface. The possible types include: <ul style="list-style-type: none"> • fe - Fast Ethernet interface • ge - Gigabit Ethernet interface • console - console interface
Oper	Operational state, up or down.
Admin	Administration state, up or down.
Last Change	The date of the last state change.

Showing Interface Statistics

To display the extended 64-bit MIB-II statistics for a specific interface or all interfaces in the CSS, use the **show mibii** command. The Enterprise ap64Stats MIB defines these statistics. To display the RFC1213 32-bit statistics, include the **-32** suffix.

To display extended MIB-II statistics for a specific interface in the CSS, enter the **show mibii** command with the interface name. To see a list of interfaces in the CSS, enter **show mibii ?**.

Table 4-7 describes the fields in the **show mibii** output.

Table 4-7 *Field Descriptions for the show mibii Command*

Field	Description
MAC	The interface's address at the protocol layer immediately below the network layer in the protocol stack. For interfaces that do not have such an address (for example, a serial line), this object should contain an octet string of zero length.
Administrative	The desired state of the interface (Enabled, Disabled, or Testing). The testing state indicates that no operational packets can be passed.
MTU	The size of the largest datagram that can be sent or received on the interface, specified in octets. For interfaces that are used for transmitting network datagrams, this is the size of the largest network datagram that can be sent on the interface.
In Octets	The total number of octets received on the interface, including framing characters.
In Unicast	The number of subnetwork-unicast packets delivered to a higher-layer protocol.
In Multicast	The number of non-unicast (for example, subnetwork-broadcast or subnetwork-multicast) packets delivered to a higher-layer protocol.
In Errors	The number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.
In Discards	The number of inbound packets that were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space.
In Unknown	The number of packets received over the interface that were discarded because of an unknown or unsupported protocol.
Last Change	The value of sysUpTime at the time the interface entered its current operational state. If the state has not changed since the time the CSS came up, the sysUptime is when the port was initialized.

Table 4-7 Field Descriptions for the `show mibii` Command (continued)

Field	Description
Operational	The current operational state of the interface (Up, Down, or Testing). The Testing state indicates that no operational packets can be passed.
Speed	An estimate of the interface's current bandwidth in bits per second. For interfaces that do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth.
Queue Len	The length of the output packet queue (in packets).
Out Octets	The total number of octets transmitted out of the interface, including framing characters.
Out Unicast	The total number of packets that higher-level protocols requested be transmitted to a subnetwork-unicast address, including those that were discarded or not sent.
Out Multicast	The total number of packets that higher-level protocols requested be transmitted to a non-unicast (for example., a subnetwork-broadcast or subnetwork-multicast) address, including those that were discarded or not sent.
Out Errors	The number of outbound packets that could not be transmitted because of errors.
Out Discards	The number of outbound packets that were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space.

To clear interface statistics, use the **clear statistics** command in SuperUser mode. For example:

```
# clear statistics
```

Showing Ethernet Interface Errors

To list the errors on an Ethernet interface, use the **show ether-errors** command and options. When required, enter the interface name as a case-sensitive unquoted text string. To see a list of interfaces, enter **show ether-errors ?**.

The command provides the following options:

- **show ether-errors** - Display the extended 64-bit statistics for errors on all Ethernet interfaces in the CSS. The Enterprise ap64Stats MIB defines these statistics.
- **show ether-errors interface name** - Display the extended 64-bit statistics for errors on a specific Ethernet interface in the CSS. The Enterprise ap64Stats MIB defines these statistics. Enter the interface name as a case-sensitive unquoted text string.
- **show ether-errors zero** - Display the Ethernet errors for all Ethernet interfaces in the CSS and reset the statistics to zero upon retrieval.
- **show ether-errors zero interface name** - Display the Ethernet errors for the specified Ethernet interface in the CSS and reset the statistics to zero upon retrieval. Enter the interface name as a case-sensitive unquoted text string.
- **show ether-errors-32** - Display the RFC1398 32-bit statistics, include the **-32** suffix.
- **show ether-errors-32 interface name** - Display the RFC1398 32-bit statistics, include the **-32** suffix. Enter the interface name as a case-sensitive unquoted text string.

Table 4-8 describes the fields in the **show ether-errors** output.

Table 4-8 Field Descriptions for the show ether-errors Command

Field	Description
Alignment	The number of frames with alignment errors (frames that do not end with a whole number of octets and have a bad CRC) received on the interface.
FCS	The number of frames received on the interface that are an integral number of octets in length but do not pass the FCS check.

Table 4-8 *Field Descriptions for the show ether-errors Command (continued)*

Field	Description
Single Collision	The number of successfully transmitted frames on the interface for which transmission is inhibited by exactly one collision.
Multiple Collisions	The number of successfully transmitted frames on the interface for which transmission is inhibited by more than one collision.
SQE Test	The number of times that the SQE TEST ERROR message is generated.
Deferred Tx	The number of frames for which the first transmission attempt on the interface is delayed because the medium is busy. The count represented by an instance of this object does not include frames involved in collisions.
Internal RX Errors	The number of frames for which reception on the interface fails due to an internal MAC sublayer receive error.
Frame too Long	The number of frames received on the interface that exceed the maximum permitted frame size.
Carrier Sense Errors	The number of times that the carrier sense condition was lost or never asserted when attempting to transmit a frame on the interface.
Internal Tx Errors	The number of frames for which transmission on the interface fails due to an internal MAC sublayer transmit error.
Excessive Collisions	The number of frames for which transmission on the interface fails due to excessive collisions.
Late Collisions	The number of times that a collision is detected on the interface later than 512 bit-times into the transmission of a packet.

Shutting Down an Interface

To shutdown an interface, use the **admin-shutdown** command in interface mode.



Caution

Shutting down an interface terminates all connections to the interface.

For example, to shut down interface e3, enter:

```
(config-if[e3]) admin-shutdown physical
```



Note

If you configure the **redundancy-phy** command on an interface and then disable the interface using the **admin-shutdown** command, the master CSS fails over to the backup CSS. To prevent the CSS from failing over when you administratively disable the interface, remove the **redundancy-phy** command by entering **no redundancy-phy** before you enter the **admin-shutdown** command on that interface.

Restarting the Interface

To restart the interface, enter the **no admin-shutdown** command. For example, to restart interface e3, enter:

```
(config-if[e3])# no admin-shutdown physical
```

Shutting Down All Interfaces

To shut down all interfaces simultaneously, use the **admin-shutdown** command at the SuperUser prompt. This command provides a quick way to shut down all physical devices in the CSS except the Console and Management ports.



Caution

Shutting down an interface terminates all connections to the interface.

To shut down all interfaces, enter:

```
# admin-shutdown
```

**Note**

To shutdown one interface, use the **admin-shutdown** command in interface mode. Refer to the “Shutting Down an Interface” section described previously in this chapter.

Restarting All Interfaces

To restart all interfaces, enter:

```
# no admin-shutdown
```

Configuring Circuits

A circuit on the CSS is a logical entity that maps IP interfaces to a logical port or group of logical ports (for example, a VLAN). Each VLAN circuit requires an IP address. Assigning an IP address to each VLAN circuit allows the CSS to route Ethernet interfaces from VLAN to VLAN.

To enter a specific circuit configuration mode, enter the **circuit** command and VLAN as shown in the following example:

```
(config)# circuit VLAN7
(config-circuit[VLAN7])#
```

**Note**

When you use the **circuit** command, enter the word ‘VLAN’ in uppercase letters and *do not* include a space between VLAN and the VLAN number (for example, VLAN7).

You can configure the following settings for a circuit:

- **router-discovery lifetime** - Configure router discovery lifetime
- **router-discovery limited-broadcast** - Transmit router discovery packets using 224.0.0.1
- **router-discovery max-advertisement-interval** - Configure router discovery maximum advertisement interval timer
- **router-discovery min-advertisement-interval** - Configure router discovery minimum advertisement interval timer

**Note**

The CSS allows you to enable router discovery and define a router discovery preference for each interface. To enable router discovery and define a preference per interface, refer to the sections “Enabling Router-Discovery” and “Configuring Router-Discovery Preference”, respectively, later in this chapter.

Configuring Router-Discovery Lifetime

To configure the maximum age in seconds that hosts remember router advertisements, use the **router-discovery lifetime** command with an integer between 0 and 9000 seconds. The default is 3 x the **max-advertisement-interval**.

For example:

```
(config-circuit[VLAN7])# router-discovery lifetime 600
```

To reset the time to the default of 3 x the **max-advertisement-interval**, enter:

```
(config-circuit[VLAN7])# no router-discovery lifetime
```

Configuring Router-Discovery Limited-Broadcast

To transmit router discovery packets using the broadcast address 255.255.255.255, use the **router-discovery limited-broadcast** command. The default is 224.0.0.1. For example:

```
(config-circuit[VLAN7])# router-discovery limited-broadcast
```

To revert to the default of 224.0.0.1, enter:

```
(config-circuit[VLAN7])# no router-discovery limited-broadcast
```

Configuring Router-Discovery Max-Advertisement-Interval

To configure router discovery maximum advertisement interval, use the **router-discovery max-advertisement-interval** command. The maximum value defines the interval between advertisements in seconds. Enter an integer from 4 to 1800. The default is 600 (10 minutes). For example:

```
(config-circuit [VLAN7])# router-discovery  
max-advertisement-interval 300
```

To restore router discovery maximum advertisement interval to the default of 600, enter:

```
(config-circuit [VLAN7])# no router-discovery  
max-advertisement-interval
```

Configuring Router-Discovery Min-Advertisement-Interval

To configure router discovery minimum advertisement interval timers, use the **router-discovery min-advertisement-interval** command. The minimum value defines the minimum interval between advertisements in seconds. Enter an integer from 0 to 1800.

The default is 0.75 x the max-advertisement-interval. If this value is greater than 0, it must be less than the maximum value.

For example:

```
(config-circuit [VLAN7])# router-discovery  
min-advertisement-interval 100
```

To reset the minimum router advertisement interval to the default of 0.75 x the maximum advertisement value, enter:

```
(config-circuit [VLAN7])# no router-discovery  
min-advertisement-interval
```

Showing Circuits

To show circuit information, use the **show circuits** command. A circuit on the CSS is a logical entity that maps IP interfaces to a logical port or group of logical ports.

This command provides the following options:

- **show circuits** - Display all circuit information for circuits that are currently up
- **show circuits all** - Display all circuit information regardless of circuit state
- **show circuit name** *circuit name* - Display circuit information for a specific circuit regardless of state

To list all circuits and their interfaces in the Up state, enter:

```
# show circuits
```

To list all circuits and their interfaces regardless of their state, enter:

```
# show circuits all
```

To list an individual circuit, enter:

```
# show circuits name VLAN5
```

Table 4-9 describes the fields in the **show circuits** output.

Table 4-9 *Field Descriptions for the show circuits Command*

Field	Description
Circuit Name	The circuit name. Note that in the show circuit output, VLANs appear as VLAN (uppercase, with no space before the VLAN number).
Circuit State	The state of the circuit. The possible states are: <ul style="list-style-type: none"> • active-ipEnabled • down-ipEnabled • active-ipDisabled • down-ipDisabled
IP Address	IP interface address.
Interface(s)	The interface associated with the circuit.
Operational Status	The operational status of the interface. The possible states are: <ul style="list-style-type: none"> • Up • Down

Configuring a Circuit IP Interface

A circuit on the CSS is a logical entity that maps IP interfaces to a logical port or group of logical ports (for example, a VLAN). Each VLAN circuit requires an IP address. Assigning an IP address to each VLAN circuit allows the CSS to route Ethernet interfaces from VLAN to VLAN.

To enter a specific circuit configuration mode, enter the **circuit** command and VLAN as shown in the following example:

```
(config)# circuit VLAN7  
(config-circuit[VLAN7])#
```

**Note**

When you use the **circuit** command, enter the word ‘VLAN’ in uppercase letters and *do not* include a space between VLAN and the VLAN number (for example, VLAN7).

The following sections describe how to define a circuit IP address.

- Configuring a Circuit IP Address
- Configuring a Circuit-IP Broadcast Address
- Configuring Circuit-IP Redirects
- Configuring Circuit-IP Unreachables
- Enabling Router-Discovery
- Configuring Router-Discovery Preference
- Enabling a Circuit IP
- Disabling a Circuit IP
- Showing IP Interfaces

Configuring a Circuit IP Address

To assign an IP address to a circuit, use the **ip address** command from the specific circuit mode. Enter the IP address and a subnet mask in CIDR bitcount notation or a mask in dot-decimal notation. The subnet mask range is 8 to 31.

For example, to configure an IP address and subnet mask for VLAN7, enter:

```
(config-circuit[VLAN7])# ip address 173.3.6.58/8
```

When you specify an IP address, the mode changes to the specific circuit-ip-VLAN-IP address as shown:

```
(config-circuit-ip[VLAN7-173.3.6.58])#
```

To remove a local IP address from a circuit, enter the following command from the circuit mode:

```
(config-circuit[VLAN7])# no ip address
```

Configuring a Circuit-IP Broadcast Address

To change the broadcast address associated with a circuit, use the **broadcast** command. If you leave the broadcast address at zero, the all-ones host is used for numbered interfaces.

The default broadcast address is an all-ones host address (for example, an IP address 173.3.6.58/24 has a broadcast address of 173.3.6.58/255). This command is available in IP configuration mode.

For example, to change the broadcast address on circuit VLAN7, enter:

```
(config-circuit-ip[VLAN7-173.3.6.58])# broadcast 0.0.0.0
```

To reset the broadcast IP address to the default all-ones host address, enter:

```
(config-circuit[VLAN7-173.3.6.58])# no broadcast
```

Configuring Circuit-IP Redirects

To enable the transmission of Internet Control Message Protocol (ICMP) redirect messages, use the **redirects** command. The default state is enabled.

For example:

```
(config-circuit-ip[VLAN7-173.3.6.58])# redirects
```

To disable the transmission of ICMP redirect messages, enter:

```
(config-circuit-ip[VLAN7-173.3.6.58])# no redirects
```

Configuring Circuit-IP Unreachables

To enable the transmission of ICMP “destination unreachable” messages, use the **unreachables** command. The default state is enabled.

For example:

```
(config-circuit-ip[VLAN7-173.3.6.58])# unreachables
```

To disable the transmission of ICMP “destination unreachable” messages, enter:

```
(config-circuit-ip[VLAN7-173.3.6.58])# no unreachables
```

Enabling Router-Discovery

To enable router discovery for an interface, use the **router-discovery** command. When enabled, router discovery transmits packets with a multicast address of 244.0.0.1. To enable an interface to transmit packets with a multicast address of 255.255.255.255, use the **router-discovery limited-broadcast** command in circuit mode (see “Configuring Router-Discovery Limited-Broadcast”, earlier in this chapter). Router discovery is disabled by default.

For example:

```
(config-circuit-ip[VLAN7-192.168.1.58])# router-discovery
```

To disable router discovery, enter:

```
(config-circuit-ip[VLAN7-192.168.1.58])# no router-discovery
```

Configuring Router-Discovery Preference

To configure the router discovery preference value, use the **router-discovery preference** command and a value to define the router preference value to advertise. The value is an integer from 0 (default) to 65535. If you use the default value, you do not need to use this command.

For example:

```
(config-circuit-ip[VLAN7-192.168.1.58])# router-discovery  
preference 100
```

To restore the router discovery preference value to the default of 0, enter:

```
(config-circuit-ip[VLAN7-192.168.1.58])# no router-discovery  
preference
```

Enabling a Circuit IP

To enable the IP interface on a circuit, use the **enable** command. The default is enable.

For example:

```
(config-circuit-ip[VLAN7-173.3.6.58])# enable
```

Disabling a Circuit IP

To disable the IP interfaces on a circuit, enter:

```
(config-circuit-ip[VLAN7-173.3.6.58])# no enable
```

Showing IP Interfaces

Use the **show ip interfaces** command to display configured IP interfaces on the CSS. The display includes the circuit state, IP address, broadcast address, Internet Control Message Protocol (ICMP) settings, and Router Discovery Program (RDP) settings.

Table 4-10 describes the fields in the **show ip interfaces** output.

Table 4-10 Field Descriptions for the show ip interfaces Command

Field	Description
Circuit Name	The name of the circuit associated with the IP interface.
State	The state of the IP interface. The possible states are: <ul style="list-style-type: none"> • active (1), the interface is up • disabled (2), the interface is disabled • noCircuit (3), the interface is waiting for an underlying circuit
IP Address	The IP address assigned to the circuit.
Network Mask	The network mask of the circuit.
Broadcast Address	The broadcast IP address associated with the IP interface. If left at zero, the all-ones host is used for numbered interfaces. 255.255.255.255 is always used for unnumbered interfaces.
Redundancy	Indicates whether or not the redundancy protocol is running on the interface. The default state is disable.
ICMP Redirect	Whether the transmission of Internet Control Message Protocol (ICMP) redirect messages is enabled or disabled. The default state is Enabled.
ICMP Unreachable	Whether the transmission of ICMP “destination unreachable” messages is enabled or disabled. The default state is enabled.
RIP	Whether the RIP is enabled or disabled.

Configuring RIP for an IP Interface

The CSS enables you to configure Routing Information Protocol (RIP) attributes on each IP interface. To configure Routing Information Protocol (RIP) parameters and run RIP on an IP interface, use the following routing commands within the specific circuit IP mode. The default mode is to send RIP version 2 (v2) and receive either RIP or RIP2.

You can configure the following routing options for each IP interface:

- **rip** - Start RIP on the IP interface
- **rip default-route** - Advertise a default route on this interface
- **rip receive** - Specify the RIP version that the IP interface receives
- **rip send** - Specify the RIP version that the IP interface sends
- **rip log** - Enables the logging of received or transmitted RIP packets

To start running RIP on an IP interface, enter:

```
(config-circuit-ip[VLAN7-192.168.1.58])# rip
```

To stop running the RIP on the interface, enter:

```
(config-circuit-ip[VLAN7-192.168.1.58])# no rip
```

Configuring RIP Default-Route

To advertise a default route on an IP interface with a specific metric, use the **rip default-route** command.

You can also specify an optional metric in the command line. The CSS uses this metric when advertising a route. Enter a number from 1 to 15. The default is 1.

For example:

```
(config-circuit-ip[VLAN7-192.168.1.58])# rip  
default-route 9
```

Configuring RIP Receive

To specify the RIP version that the interface receives, use the **rip receive** command. The **rip receive** options are:

- **rip receive both** - Receive both RIP version 1 and RIP version 2 (default)
- **rip receive none** - Receive no RIP packets
- **rip receive v1** - Receive RIP version 1 packets only
- **rip receive v2** - Receive RIP version 2 packets only

For example:

```
(config-circuit-ip[VLAN7-192.168.1.58])# rip receive both
```

Configuring RIP Send

To specify the RIP version that the interface sends, use the **rip send** command. The **rip send** options are:

- **rip send none** - Send no RIP packets
- **rip send v1** - Send RIP version 1 packets only
- **rip send v2** - Send RIP version 2 packets only (default)

For example:

```
(config-circuit-ip[VLAN7-192.168.1.58])# rip send v1
```

Configuring RIP Packet Logging

To enable the logging of received or transmitted RIP packets on the interface, use the **rip log** command. Use the **no** form of this command to disable logging (default setting). The **rip log** options are:

- **rip log rx** - Specifies that the CSS logs RIP packets received on the interface
- **rip log tx** - Specifies that the CSS logs RIP packets transmitted on the interface

For example:

```
(config-circuit-ip[VLAN7-192.168.1.58])# rip log rx
```

Showing RIP Configurations

To show a RIP configuration for one IP address or all IP addresses configured in the CSS, use the **show rip** command. The command provides the following options:

- **show rip** - Displays RIP configurations for all interfaces (including the logging of RIP packets)
- **show rip ip_address** - Displays a single RIP interface entry
- **show rip globals** - Displays RIP global statistics
- **show rip statistics** - Displays RIP interface statistics for all interfaces
- **show rip statistics ip_address** - Displays RIP interface statistics for a specific interface

Table 4-11 describes the fields in the **show rip** output.

Table 4-11 Field Descriptions for the show rip Command

Field	Description
IP Address	The advertised RIP interface address.
State	The operational state of the RIP interface.
RIP Send	The RIP version that the interface sends. The possible field values are: <ul style="list-style-type: none"> • none, do not send RIP packets • RIPv1, send RIP version 1 packets only • RIPv2, send RIP version 2 packets only (default)
RIP Recv	The RIP version that the interface receives. The possible values are: <ul style="list-style-type: none"> • both, receiving both version 1 and version 2 (default) • none, receiving no RIP packets • RIPv1, receiving RIP version 1 packets only • RIPv2, receiving RIP version 2 packets only
Default Metric	The default metric used when advertising the RIP interface.
Tx Log	The setting for the logging of RIP packet transmissions (enabled or disabled). The default setting is disabled.
Rx Log	The setting for the logging of RIP packet received (enabled or disabled). The default setting is disabled.

To display global RIP statistics, enter:

```
# show rip globals
```

Table 4-12 describes the fields in the **show rip globals** output.

Table 4-12 Field Descriptions for the show rip globals Command

Field	Description
RIP Route Changes	The global number of route changes made to the IP route database by RIP
RIP Query Responses	The global number of query responses sent to RIP query from other systems

To display the RIP interface statistics for all RIP interface entries, enter:

```
# show rip statistics
```

Table 4-13 describes the fields in the **show rip statistics** output.

Table 4-13 Field Descriptions for the show rip statistics Command

Field	Description
System Route Changes	The global number of route changes made to the IP route database by RIP
System Global Query Responses	The global number of query responses sent to RIP query from other systems
IP Address	The RIP interface IP address
Triggered Updates Sent	The number of triggered RIP updates sent by the interface
Bad Packets Received	The number of bad RIP response packets received by the interface
Bad Routes Received	The number of bad routes in valid RIP packets received by the interface

Where to Go Next

For information on creating and configuring services, refer to Chapter 5, Configuring Services.

