Cisco Content Services Switch Administration Guide

Software Version 7.20
March 2003

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Customer Order Number: DOC-7813885=
Text Part Number: 78-13885-05

Cisco Content Services Switch Administration Guide

CONTENTS

CHAPTER 11

FIGURES

TABLES

Preface

This guide provides instructions for the administration of the Cisco 11500 Series Content Services Switches (CSS). It describes how to perform administration tasks on the CSS, including logging in to the CSS, configuring CSS Ethernet interface ports, configuring network protocols, upgrading your CSS software, and so on. Information in this guide applies to all CSS models except where noted.

For information on basic CSS configuration, refer to the *Content Services Switch Basic Configuration Guide*. For information on configuring advanced features, refer to the *Content Services Switch Advanced Configuration Guide*.

The CSS software is available in a Standard or optional Enhanced feature set. The Enhanced feature set contains all of the Standard feature set and also includes Network Address Translation (NAT) Peering, Domain Name Service (DNS), Demand-Based Content Replication (Dynamic Hot Content Overflow), Content Staging and Replication, and Network Proximity DNS. Proximity Database and Secure Management, which includes Secure Shell Host and SSL strong encryption for the Device Management software, are optional features.

Note You must enter a Standard software license key when you boot the CSS for the first time. Refer to Chapter 1, Booting, Logging In, and Getting Started for details about activating a CSS software option.

This preface describes the following topics:

- Audience
- How to Use This Guide
- Related Documentation
- Symbols and Conventions
- Obtaining Documentation
- Obtaining Technical Assistance
- Obtaining Additional Publications and Information

Audience

This guide is intended for the following trained and qualified service personnel who are responsible for configuring the CSS:

- Web master
- System administrator
- System operator

How to Use This Guide

This guide is organized as follows:

Chapter	Description
Chapter 1, Booting, Logging In, and Getting Started	Power-on and boot the CSS for the first time, log in to the CSS, and boot the CSS on a routine basis.
Chapter 2, Configuring CSS Basics	Basic configuration of the CSS, including the username and password, Ethernet management port, static IP routes, and the date and time.

Chapter	Description
Chapter 3, Managing the CSS Software	Copy the running-configuration and startup-configuration files, specify file storage locations for a two-disk Cisco 11500 series CSS, and unpack and remove an ArrowPoint Distribution Image (ADI). This chapter also includes an overview on the CSS system software.
Chapter 4, Specifying the CSS Boot Configuration	Set the primary and secondary boot configuration for the CSS.
Chapter 5, Configuring Interfaces and Circuits	Configure the CSS interface ports and circuits for operation.
Chapter 6, Configuring CSS Network Protocols	Configure Domain Name Service (DNS), Address Resolution Protocol (ARP), Routing Information Protocol (RIP), Internet Protocol (IP), and spanning-tree bridging, and Dynamic Host Configuration Protocol (DHCP).
Chapter 7, Configuring Open Shortest Path First (OSPF)	Configure OSPF routing protocol.
Chapter 8, Using the CSS Logging Features	Configure logging for the CSS. This chapter also provides information displaying and interpreting log messages.
Chapter 9, Configuring Flow and Port Mapping Parameters	Configure flow parameters for the CSS, including connections for TCP or UDP ports, flow inactivity timeout values, and port mapping
Chapter 10, Configuring User Profiles	Configure user profiles in the default-profile file.
Chapter 11, Configuring CSS Remote Access Methods	Configure CSS remote access methods, including the Secure Shell Daemon (SSH) protocol, the Remote Authentication Dial-In User Service (RADIUS) protocol, and the Terminal Access Controller Access Control System (TACACS+) protocol.

Chapter	Description
Chapter 12, Configuring Simple Network Management Protocol (SNMP)	Configure SNMP on the CSS. This chapter also includes a summary of all CSS Enterprise Management Information Base (MIB) objects.
Chapter 13, Configuring Remote Monitoring (RMON)	Configure RMON on the CSS.
Appendix A, Upgrading Your CSS Software	Upgrade your CSS software manually or use the upgrade script.
Appendix B, Using the Offline Diagnostic Monitor Menu	Information on using the Offline Diagnostic Monitor (Offline DM) menu.
Appendix C, Troubleshooting the Boot Process	Troubleshoot the boot process for the Cisco 11500 series CSS.

Related Documentation

In addition to this document, the CSS documentation set includes the following:

Document Title	Description
Release Note for the Cisco 11500 Series Content Services Switch	This release note provides information on operating considerations, caveats, and command line interface (CLI) commands for the Cisco 11500 series CSS.
Cisco 11500 Series Content Services Switch Hardware Installation Guide	This guide provides information for installing, cabling, and powering the Cisco 11500 series CSS. In addition, this guide provides information about CSS specifications, cable pinouts, and hardware troubleshooting.

Document Title	Description
Content Services Switch Basic Configuration Guide	This guide describes how to perform basic CSS configuration tasks, including: • Services • Owners • Content rules • Sticky parameters • Source groups, access control lists (ACLs), Extension Qualifier Lists (EQLs), Uniform Resource Locator Qualifier Lists (URQLs), Network Qualifier Lists (NQLs), and Domain Qualifier Lists (DQLs) • HTTP header load balancing • Content caching
Content Services Switch Advanced Configuration Guide	This guide describes how to perform advanced CSS configuration tasks, including: • Domain Name Service (DNS) • DNS Sticky • Content Routing Agent • Client-Side Accelerator • Network proximity • VIP and virtual interface redundancy • Box-to-box redundancy • Demand-based content replication and content staging and replication • Secure Socket Layer (SSL) termination with the SSL Acceleration Module • Firewall load balancing • CSS scripting language • XML documents to configure the CSS

Document Title	Description
Content Services Switch Command Reference	Provides an alphabetical list of all CLI commands including syntax, options, and related commands.
Content Services Switch Device Management User's Guide	Provides an overview of using the Device Management user interface, an HTML-based Web-based application that you use to configure and manage your CSS.

Symbols and Conventions

This guide uses the following symbols and conventions to identify different types of information.

Caution A caution means that a specific action you take could cause a loss of data or adversely impact use of the equipment.

Warning **A warning describes an action that could cause you physical harm or damage the equipment.**

Note A note provides important related information, reminders, and recommendations.

Bold text indicates a command in a paragraph.

`Courier text` indicates text that appears on a command line, including the CLI prompt.

`Courier bold text` indicates commands and text you enter in a command line.

Italic text indicates the first occurrence of a new term, book title, emphasized text, and variables for which you supply values.

1. A numbered list indicates that the order of the list items is important.
    a. An alphabetical list indicates that the order of the secondary list items is important.

- A bulleted list indicates that the order of the list topics is unimportant.
    - An indented list indicates that the order of the list subtopics is unimportant.

Obtaining Documentation

Cisco provides several ways to obtain documentation, technical assistance, and other technical resources. These sections explain how to obtain technical information from Cisco Systems.

Cisco.com

You can access the most current Cisco documentation on the World Wide Web at this URL:

http://www.cisco.com/univercd/home/home.htm

You can access the Cisco website at this URL:

http://www.cisco.com

International Cisco web sites can be accessed from this URL:

http://www.cisco.com/public/countries_languages.shtml

Documentation CD-ROM

Cisco documentation and additional literature are available in a Cisco Documentation CD-ROM package, which may have shipped with your product. The Documentation CD-ROM is updated monthly and may be more current than printed documentation. The CD-ROM package is available as a single unit or through an annual subscription.

Registered Cisco.com users can order the Documentation CD-ROM (product number DOC-CONDOCCD=) through the online Subscription Store:

http://www.cisco.com/go/subscription

Ordering Documentation

You can find instructions for ordering documentation at this URL:

http://www.cisco.com/univercd/cc/td/doc/es_inpck/pdi.htm

You can order Cisco documentation in these ways:

- Registered Cisco.com users (Cisco direct customers) can order Cisco product documentation from the Networking Products MarketPlace:

  http://www.cisco.com/en/US/partner/ordering/index.shtml

- Registered Cisco.com users can order the Documentation CD-ROM (Customer Order Number DOC-CONDOCCD=) through the online Subscription Store:

  http://www.cisco.com/go/subscription

- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco Systems Corporate Headquarters (California, U.S.A.) at 408 526-7208 or, elsewhere in North America, by calling 800 553-NETS (6387).

Documentation Feedback

You can submit comments electronically on Cisco.com. On the Cisco Documentation home page, click **Feedback** at the top of the page.

You can e-mail your comments to bug-doc@cisco.com.

You can submit your comments by mail by using the response card behind the front cover of your document or by writing to the following address:

Cisco Systems
Attn: Customer Document Ordering
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

Obtaining Technical Assistance

Cisco provides Cisco.com, which includes the Cisco Technical Assistance Center (TAC) Website, as a starting point for all technical assistance. Customers and partners can obtain online documentation, troubleshooting tips, and sample configurations from the Cisco TAC website. Cisco.com registered users have complete access to the technical support resources on the Cisco TAC website, including TAC tools and utilities.

Cisco.com

Cisco.com offers a suite of interactive, networked services that let you access Cisco information, networking solutions, services, programs, and resources at any time, from anywhere in the world.

Cisco.com provides a broad range of features and services to help you with these tasks:

- Streamline business processes and improve productivity
- Resolve technical issues with online support
- Download and test software packages
- Order Cisco learning materials and merchandise
- Register for online skill assessment, training, and certification programs

To obtain customized information and service, you can self-register on Cisco.com at this URL:

http://www.cisco.com

Technical Assistance Center

The Cisco TAC is available to all customers who need technical assistance with a Cisco product, technology, or solution. Two levels of support are available: the Cisco TAC website and the Cisco TAC Escalation Center. The avenue of support that you choose depends on the priority of the problem and the conditions stated in service contracts, when applicable.

We categorize Cisco TAC inquiries according to urgency:

- Priority level 4 (P4)—You need information or assistance concerning Cisco product capabilities, product installation, or basic product configuration.
- Priority level 3 (P3)—Your network performance is degraded. Network functionality is noticeably impaired, but most business operations continue.
- Priority level 2 (P2)—Your production network is severely degraded, affecting significant aspects of business operations. No workaround is available.
- Priority level 1 (P1)—Your production network is down, and a critical impact to business operations will occur if service is not restored quickly. No workaround is available.

Cisco TAC Website

You can use the Cisco TAC website to resolve P3 and P4 issues yourself, saving both cost and time. The site provides around-the-clock access to online tools, knowledge bases, and software. To access the Cisco TAC website, go to this URL:

http://www.cisco.com/tac

All customers, partners, and resellers who have a valid Cisco service contract have complete access to the technical support resources on the Cisco TAC website. Some services on the Cisco TAC website require a Cisco.com login ID and password. If you have a valid service contract but do not have a login ID or password, go to this URL to register:

http://tools.cisco.com/RPF/register/register.do

If you are a Cisco.com registered user, and you cannot resolve your technical issues by using the Cisco TAC website, you can open a case online at this URL:

http://www.cisco.com/en/US/support/index.html

If you have Internet access, we recommend that you open P3 and P4 cases through the Cisco TAC website so that you can describe the situation in your own words and attach any necessary files.

Cisco TAC Escalation Center

The Cisco TAC Escalation Center addresses priority level 1 or priority level 2 issues. These classifications are assigned when severe network degradation significantly impacts business operations. When you contact the TAC Escalation Center with a P1 or P2 problem, a Cisco TAC engineer automatically opens a case.

To obtain a directory of toll-free Cisco TAC telephone numbers for your country, go to this URL:

http://www.cisco.com/warp/public/687/Directory/DirTAC.shtml

Before calling, please check with your network operations center to determine the level of Cisco support services to which your company is entitled: for example, SMARTnet, SMARTnet Onsite, or Network Supported Accounts (NSA). When you call the center, please have available your service agreement number and your product serial number.

Obtaining Additional Publications and Information

Information about Cisco products, technologies, and network solutions is available from various online and printed sources.

- The *Cisco Product Catalog* describes the networking products offered by Cisco Systems as well as ordering and customer support services. Access the *Cisco Product Catalog* at this URL:

  http://www.cisco.com/en/US/products/products_catalog_links_launch.html

- Cisco Press publishes a wide range of networking publications. Cisco suggests these titles for new and experienced users: *Internetworking Terms and Acronyms Dictionary, Internetworking Technology Handbook, Internetworking Troubleshooting Guide,* and the *Internetworking Design Guide.* For current Cisco Press titles and other information, go to Cisco Press online at this URL:

  http://www.ciscopress.com

- *Packet* magazine is the Cisco monthly periodical that provides industry professionals with the latest information about the field of networking. You can access *Packet* magazine at this URL:

  http://www.cisco.com/en/US/about/ac123/ac114/about_cisco_packet_magazine.html

- *iQ Magazine* is the Cisco monthly periodical that provides business leaders and decision makers with the latest information about the networking industry. You can access *iQ Magazine* at this URL:

  http://business.cisco.com/prod/tree.taf%3fasset_id=44699&public_view=true&kbns=1.html

- *Internet Protocol Journal* is a quarterly journal published by Cisco Systems for engineering professionals involved in the design, development, and operation of public and private internets and intranets. You can access the *Internet Protocol Journal* at this URL:

  http://www.cisco.com/en/US/about/ac123/ac147/about_cisco_the_internet_protocol_journal.html

- Training—Cisco offers world-class networking training, with current offerings in network training listed at this URL:

  http://www.cisco.com/en/US/learning/le31/learning_recommended_training_list.html

CHAPTER 1

Booting, Logging In, and Getting Started

This chapter describes how to boot the CSS for the first time and on a routine basis, and how to log in. It also covers using the configuration script, which initiates automatically when you log in *and* the CSS does not detect an existing startup-config file. Information in this chapter applies to all CSS models, except where noted.

This chapter contains the following major sections:

- Booting the CSS for the First Time
- Booting the CSS on a Routine Basis
- Logging in to the CSS
- Using the Configuration Script
- Rebooting the CSS
- Shutting Down the CSS

Booting the CSS for the First Time

Upon bootup, the CSS initially:

- Performs hardware initialization and power-on diagnostics (as described in the "Booting the CSS on a Routine Basis" section)
- Prompts you to:
  - Enter the Standard software license key
  - Configure the IP address, subnet mask, and default gateway for the Ethernet management port, used for CSS configuration and Ethernet management only; this port does not route traffic
  - Change the default administrative login name (**admin**) and password (**system**)
  - Password protect the Offline Diagnostic Monitor (Offline DM) menu

This sections includes the following procedures:

- Entering Your License Key
- Configuring the Ethernet Management Port
- Changing the Default Username and Password
- Password Protecting the Offline DM Menu

Entering Your License Key

When the CSS completes hardware initialization and power-on diagnostics, you must enter a valid Standard license key for the CSS software. The CSS does not require you to enter this key on subsequent startups.

Locate your Standard software license key inside the CSS accessory kit. If you cannot locate the software license key, call the Cisco Technical Assistance Center (TAC) toll free, 24 hours a day, 7 days a week at 1-800-553-2447 or 1-408-526-7209. You can also e-mail TAC at tac@cisco.com.

The CSS prompts you to accept the license agreement. You must accept the license agreement or you cannot log in to the CSS.

The CSS prompts you to enter your Standard software license key, as follows:

```
Enter Software License Key: xxxxxxxxxxxx
```

If you enter:

- A valid license number, the CSS prompts you to enter an IP address for the Ethernet management port.
- An invalid license number, the CSS redisplays the license prompt until you enter a valid number. If you do not enter a valid license number, you cannot log in to the CSS.

If, during the initial CSS order placement, you purchased the Enhanced feature set, the Secure Management option (which includes Secure Shell Host and SSL strong encryption for the Device Management software), or the Proximity Database software option, locate the software Claim Certificate in the accessory kit. Follow the instructions on the Claim Certificate to obtain a license key from Cisco Systems for the additional software feature.

After you receive the software license key, use the **license** command to enter the license key. At the prompt, enter the license key. For example, enter:

```
# license
Enter Software License Key (q to quit):
```

Note After you enter the software license key for the Proximity Database software option, you must reboot the CSS for the license key to take effect.

Configuring the Ethernet Management Port

Once you enter a valid license key at the boot prompt, the CSS displays the following message and prompt:

```
Use the Ethernet management port IP address to access the Content
Services Switch for configuration and management only. This port does
not route traffic and is not associated with VLAN circuits.

The current address setting (0.0.0.0) disables the Ethernet Management
port.

Do you wish to configure a valid address for the Ethernet management
port [y/n]?
```

Enter one of the following:

- **y** to configure an IP address, subnet mask, and default gateway for the Ethernet management port. The CSS prompts you for an IP address, a subnet mask, then a default gateway. You must enter a valid IP address or the CSS repeats the prompt until you do.

  ```
  Enter IP Address [0.0.0.0]:
  Enter Subnet Mask [0.0.0.0]:
  Enter Default Gateway [0.0.0.0]:
  ```

Note The Ethernet management port IP address must be on a different subnet from any other CSS VLAN circuit subnet. If you do not make this IP address unique, you will not be able to access the port.

- **n** to accept the default IP address (0.0.0.0), subnet mask (0.0.0.0), and gateway (0.0.0.0) and to disable the port. The CSS does not prompt you for an IP address, subnet mask, and default gateway.

The Ethernet management port default IP address of 0.0.0.0 disables the Ethernet management port. To enable the Ethernet management port, specify the **ip address** command in boot mode (see Chapter 2, Configuring CSS Basics) or use the Offline DM menu (see Appendix B, Using the Offline Diagnostic Monitor Menu).

Changing the Default Username and Password

The CSS allows you to change the default username and password. We recommend that you change them to safeguard the CSS against unauthorized logins.

```
Access to this device is allowed using the default username and
password. For enhanced security we recommend that you change the
defaults. Do you want to change the defaults now (yes,no):
```

Enter one of the following:

- **yes** to change the username and password. The CSS prompts you for the following information and password confirmation.

  ```
  Enter <administrator> username:
  Enter <administrator> password:
  Confirm <administrator> password:
  ```

- **no** to keep the default username and password.

To change the default username and password from the CLI, refer to Chapter 3, Managing the CSS Software for details.

Password Protecting the Offline DM Menu

The CSS prompts you to password-protect the Offline DM menu.

```
Set Password Protection for Offline Diagnostic Monitor menu (yes,no)
```

⚠ **Caution** Use care when password protecting the Offline DM menu and ensure that you write down the new password. If you lose the new password, it cannot be recovered and you will be unable to access the Offline DM Main menu. The only solution, at that point, is to contact the Cisco Technical Assistance Center (TAC) at 1-800-553-2447 or 1-408-526-7209. You can also e-mail TAC at tac@cisco.com.

Enter one of the following:

- **yes** to password protect the Offline DM menu. When you password protect the Offline DM menu, you need to enter the administrative username and password each time you access the menu.

  ```
  The administrative username and password are required to access
  the Offline Diagnostic Monitor menu.
  Initializing the disk..........OK
  ```

  Refer to Appendix B, Using the Offline Diagnostic Monitor Menu, for information on the Offline DM menu options.

- **no** to disable password protection on the Offline DM menu.

The CSS prompts you to access the Offline DM menu.

```
Would you like to access the Offline Diagnostic Monitor? (Y <cr>)
```

Enter **y** to access the Offline DM menu. If you do not wish to access the Offline DM menu after seeing this message, wait for the CSS to boot.

Booting the CSS on a Routine Basis

When you power up a CSS, the boot process:

- Displays the software version and build number
- Performs hardware initialization and power-on self tests
- Provides access to the Offline DM menu
- Prompts you to log in to the CSS

The duration of the boot process depends on the CSS startup configuration and, with the CSS 11503 and CSS 11506, the number of modules in the chassis.

When you boot the CSS, it initializes the hardware and performs power-on self tests. The CSS displays the following messages (shown for the CSS 11503 and CSS 11506):

```
Locked boot flash.
Validating operational boot flash, please wait...
Operational boot flash valid. Jumping to operational boot flash.
Copyright 2002(c), Cisco Systems, Inc.

Operational boot flash.
Attaching interrupt handlers...Done.
Master SCM.
Built Jun 22 2002 @ 15:14:20
Version x.xx Build xx
```

Note After the CSS begins to boot (approximately 15 seconds) the CSS allows you to access the Offline DM menu. The Offline DM Main menu allows you to set the boot configuration, display the boot configuration, select Advanced Options, or reboot the system. Refer to Appendix B, Using the Offline Diagnostic Monitor Menu for detailed information on using Offline DM.

The hardware then goes through a series of power-on self tests. The asterisks that appear indicate the completion of each test.

```
Press <ESC> to enter the Diagnostic Monitor
* * * * * * * * * * * * * *...
Ran  1 times, x tests. Detected 0 errors.
```

During the power-on self tests, the Status LEDs blink and change color to indicate the stages of the boot process. The left Status LED is bicolor, green or red. The right Status LED is amber.

The Ethernet connectors on the CSS 11501 and the 8- and 16-port Fast Ethernet Modules on the CSS 11503 or CSS 11506 do not contain Status LEDs. Each Ethernet connector has Link and Duplex LEDs to indicate the state of the connection.

Table 1-1 defines the boot states and the blinking patterns of the Status LEDs.

Table 1-1 Status LEDs Boot Definitions

State Sequence		LED Color	LED State
1.	The CSS powers up, flash scans, and does a power-on self test.	Amber	Fast blink
	The CSS powers on and a self test detects an error.	Red	Solid
2.	The CSS 11501 or a module in the CSS 11503 or CSS 11506 is off line and active.	Amber	Slow blink
3.	The CSS 11501 or a module in the CSS 11503 or CSS 11506 is online and not active. In the CSS 11506, a passive SCM LED remains in this state and color.	Amber	Solid

Table 1-1 Status LEDs Boot Definitions (continued)

State Sequence		LED Color	LED State
4.	The CSS 11501, or a module in the CSS 11503 or CSS 11506, is on line and active.	Green	Solid
	The CSS 11501 or a module in the CSS 11503 or CSS 11506 (except a Fast Ethernet Module) failed. In the CSS 11503 or 11506, if: • A Fast Ethernet Module fails, all of the Link and Duplex LEDs blink simultaneously. • The master SCM in slot 1 detects a module failure, its Status LED is green and blinks slowly. • The master SCM in slot 1 fails, the CSS does not boot unless there is a passive SCM in slot 2.	Red	Blinking
5.	Disk activity	Green	Variable blinking

If an error occurs during a power-on self-test, the console displays an error message, increments the detected error counter, and continues to the next test until the CSS completes all of the power-on self tests. Refer to Chapter 8, Using the CSS Logging Features and Appendix C, Troubleshooting the Boot Process for more information on boot errors and messages.

Logging in to the CSS

After the CSS completes the boot process, it displays the login banner, copyright, and login prompt.

When a startup-config file is present, the CSS displays the message: `Press CTRL-C to abort running the startup-config`

Note If the CSS does not detect an existing startup-config file, the CSS automatically initiates the configuration script (see the "Using the Configuration Script" section). The configuration script prompts you to enter configuration information. Subsequent logins to the CSS do not start the configuration script.

If you abort running the startup-config file, the CSS does not use the existing startup-config file. Aborting the use of the startup-config file enables you to log in and reconfigure the CSS to create a new running-config file. Use this feature if you misconfigure your startup-config file and the CSS becomes unusable.

When you log in from:

- A console, the CSS displays the message: `Press any key to log in.`
- A Telnet session, the message is not displayed.

The CSS prompts you to enter a username and password, as follows:

```
User Access Verification
Username:
Password
```

If you connect a console to the CSS after the CSS boots, your screen will be blank. Press **Enter** to display the username and password prompts.

To initially log in to the CSS, enter the default user name **admin** and the default password **system** as lowercase text, or enter the administrative username and password you configured during the boot process. For security, the CSS does not display the password. The default username **admin** enables you to log in with SuperUser status.

If you have not changed the default administrative username and password, we ecommend that you change them to safeguard the CSS against unauthorized logins. To change the default username and password from the CLI, refer to Chapter 2, Configuring CSS Basics.

Using the Configuration Script

When you log in to the CSS and it does not detect an existing startup-config file, the CSS automatically initiates the configuration script. During the running of the configuration script, the CSS prompts you to enter the following information:

- IP address and subnet mask for circuit VLAN1 (all interfaces are assigned to VLAN1 by default)
- IP address for the default gateway
- IP addresses for the servers
- Virtual IP address (VIP) for the content rule

Based on your entries, the configuration script allows you to create services, owners, and content rules. For background information on configuring services, owners, and content rules, refer to the *Cisco Content Services Switch Basic Configuration Guide*.

To accept the script default values, press the **Enter** key at the prompts shown in the configuration script. To quit the script, enter **q** at any prompt. If you quit running the script, you may proceed to Chapter 2, Configuring CSS Basics to continue the initial setup of the CSS.

Note You may also initiate the configuration script manually by entering the **script play setup** command.

To clear an existing running-config file, use the **clear running-config** command from SuperUser mode. To clear an existing startup-config file, use the **clear startup-config** command from SuperUser mode.

The following example illustrates the configuration script including:

- **Bold** text to indicate user entry examples
- Explanations to help you use the script

```
##############################################
#Setup Script for the Content Services Switch#
##############################################

Checking for Existing Config...

No startup-config was found, continue with the setup script [y/n]? y

Note: Pressing “q” after any prompt quits setup. Pressing <CR> after
any [y/n] defaults to “y”.

Warning: All circuit VLAN IP addresses must be on a different subnet
than the Ethernet Mgt port IP address. The existing Ethernet Mgt port
IP address is: 10.0.4.251

Add an IP address to VLAN1: [default = 192.168.10.1] 192.168.3.6

Add an IP subnet mask to VLAN1: [default = 255.255.255.0]

Warning: The default gateway IP address must be on the same subnet as
VLAN1. VLAN1 IP address is: 192.168.3.6

Add IP address for default gateway: [default = 192.168.3.2]
192.168.3.3

Pinging the default gateway: 100% Success.

Which feature do you want to configure?

[1] Layer3 load balancing
[2] Layer5 load balancing
[3] Proxy cache
[4] Transparent cache
[5] Exit script
```

Table 1-2 describes each Configuration Script menu item.

Table 1-2 Configuration Script Menu Options

Menu Option	Function
Layer3 Load Balancing	Configure Layer 3 load balancing to enable the CSS to use a Virtual IP address (VIP) to load balance Web traffic to Web servers based on IP addresses
Layer5 Load Balancing	Configure Layer 5 load balancing to enable the CSS to use a VIP address to load balance Web traffic to Web servers based on URLs.
Proxy Cache	Configure proxy cache to enables the CSS to use a Virtual IP address (VIP) to load balance Web traffic to proxy cache servers based on domain name.
Transparent Cache	Configure transparent cache to enable the CSS to redirect cacheable HTTP traffic to transparent cache servers based on IP address and port (80).
Exit Script	Exit from the script and save the information you entered to the CSS running-config file. The CSS displays the running-config file.

Refer to the following sections for details about each item in the Configuration Script menu:

- Configuring Layer 3 Load Balancing
- Configuring Layer 5 Load Balancing
- Configuring Proxy Cache
- Configuring Transparent Cache

Configuring Layer 3 Load Balancing

A Layer 3 load-balancing configuration enables the CSS to use a Virtual IP address (VIP) to load balance Web traffic to Web servers based on IP addresses.

When you select Layer 3 load balancing, the script automatically:

- Creates an owner (L3_Owner)
- Creates a Layer 3 content rule (L3_Rule) and defines ArrowPoint Content Awareness (ACA) as the load balance method
- Activates the services
- Activates the content rule
- Saves the running configuration to the startup-config file

The script prompts you to configure:

- Service name (default name is Server1)
- Service IP address
- VIP for the content rule

To configure Layer 3 load balancing, enter **1** at the Configuration Script menu.

```
Which feature do you want to configure?

[1] Layer3 load balancing
[2] Layer5 load balancing
[3] Proxy cache
[4] Transparent cache

Enter the number for the feature you want to configure: 1
```

To accept the script default values, press the **Enter** key at the prompts.

```
Creating Layer3 load balancing

Enter service name: [default = Server1]

Enter service IP address: [default = 192.168.10.3] 192.168.3.58

Create another service? [y/n]? y

Enter service name: [default = Server2]

Enter service IP address: [default = 192.168.10.3] 192.168.3.59
```

```
Create another service? [y/n]? n

Enter Virtual IP address for L3_Rule: [default = 192.168.10.4]
192.168.3.6
```

After you specify the configuration, the script automatically:

- Displays the running-config file
- Saves the running configuration to the startup-config file

```
Showing the Running Config

!Generated MAR 6 17:53:49

!**************** GLOBAL *****************
ip route 0.0.0.0 0.0.0.0 192.168.3.3
!**************** CIRCUIT ****************
circuit VLAN1
ip address 192.168.3.6 255.255.255.0
!**************** SERVICE ****************
service Server1
    ip address 192.168.3.58
    active
service Server2
    ip address 192.168.3.59
    active
!**************** OWNER ******************
owner L3_Owner
    content L3_Rule
    add service Server1
    add service Server2
    vip address 192.168.3.6
    balance aca
    active
##########################################
##    Setup Completed Successfully!!!   ##
##########################################
```

Configuring Layer 5 Load Balancing

A Layer 5 load-balancing configuration enables the CSS to use a VIP address to load balance Web traffic to Web servers based on URLs.

When you select Layer 5 load balancing, the script automatically:

- Creates an owner (L5_Owner)
- Creates a Layer 3 content rule (L3_Rule)
- Creates a Layer 5 content rule (L5_Rule) and defines:
  - Protocol TCP
  - Port 80
  - URL "/*"
  - Load balance method as ACA
- Activates the services
- Activates the content rule
- Saves the running configuration to the startup-config file

The script prompts you to configure:

- Service name (default name is Server1)
- VIP for the content rule

To configure Layer 5 load balancing, enter **2** at the Configuration Script menu..

```
Which feature do you want to configure?

[1] Layer3 load balancing
[2] Layer5 load balancing
[3] Proxy cache
[4] Transparent cache

Enter the number for the feature you want to configure: 2
```

To accept the script default values, press the **Enter** key at the prompts.

```
Creating Layer5 load balancing

Enter service name: [default= Server1]

Enter service IP address: [default = 192.168.10.3] 192.168.3.58
```

```
Create another service? [y/n]? n

Enter Virtual IP address for L5_Rule: [default = 192.168.10.4]
192.168.3.8
```

After you specify the configuration, the script automatically:

- Displays the running-config file
- Saves the running configuration to the startup-config file

```
Showing the Running Config

!Generated MAR 6 17:53:49

!**************** GLOBAL *****************
ip route 0.0.0.0 0.0.0.0 192.168.3.3
!**************** CIRCUIT ****************
circuit VLAN1
ip address 192.168.3.6 255.255.255.0
!**************** SERVICE ****************
service Server1
    ip address 192.168.3.58
    active
!**************** OWNER ******************
owner L5_Owner
content L3_Rule
    add service Server1
    vip address 192.168.3.8
    balance aca
    active
content L5_Rule
    add service Server1
    vip address 192.168.3.8
    protocol tcp
    port 80
    url "/*"
    balance aca
    active
##########################################
##    Setup Completed Successfully!!!   ##
##########################################
```

Configuring Proxy Cache

A proxy cache configuration enables the CSS to use a Virtual IP address (VIP) to load balance Web traffic to proxy cache servers based on domain name.

When you select Proxy Cache, the script automatically:

- Creates an owner (Proxy_Owner)
- Creates a content rule (Proxy_Rule) and defines:
  - Service type as proxy-cache
  - Protocol TCP
  - Port 8080
  - URL "/*"
  - Load balance method as domain
  - Application type HTTP
- Activates the services
- Activates the content rule

The script prompts you to configure:

- Service name (default name is Proxy_Cache1)
- VIP for the content rule

To configure a proxy cache configuration, enter **3** at the Configuration Script menu.

```
Which feature do you want to configure?

[1] Layer3 load balancing
[2] Layer5 load balancing
[3] Proxy cache
[4] Transparent cache

Enter the number for the feature you want to configure: 3
```

To accept the script default values, press the **Enter** key at the prompts.

```
Creating Proxy Cache Configuration

Enter service name: [default=Proxy_Cache1]

Enter service IP address: [default = 192.168.10.3] 192.168.3.60
```

```
Create another service? [y/n]? n

Enter Virtual IP address for Proxy_Rule: [default = 192.168.10.4]
192.168.3.9
```

After you specify the configuration, the script automatically:

- Displays the running-config file
- Saves the running configuration to the startup-config file

```
Showing the Running Config
!Generated MAR 6 17:53:49
!**************** GLOBAL *****************
ip route 0.0.0.0 0.0.0.0 192.168.3.3
!**************** CIRCUIT ****************
circuit VLAN1
ip address 192.168.3.6 255.255.255.0
!**************** SERVICE ****************
service Proxy_Cache1
    ip address 192.168.3.60
    type proxy-cache
    port 8080
    protocol tcp
    active
!**************** OWNER ******************
owner Proxy_Owner
content Proxy_Rule
    add service Proxy_Cache1
    vip address 192.168.3.9
    port 8080
    protocol tcp
    url "/*"
    balance domain
    application http
    active

##########################################
##    Setup Completed Successfully!!!   ##
##########################################
```

Configuring Transparent Cache

A transparent cache configuration enables the CSS to redirect cacheable HTTP traffic to transparent cache servers based on IP address and port (80). The CSS directs non-cacheable HTTP traffic to the origin servers.

When you select Transparent Cache, the script automatically:

- Creates an owner (Transparent_Owner)
- Creates a content rule (Transparent_Rule) and defines:
  - Service type as transparent-cache
  - Protocol TCP
  - Port 80
  - Extension Qualifier List (EQL) named *Cacheable* that contains the file types displayed in the sample running-config file
  - URL "/*" eql cacheable
  - Load balance method as domain
  - Failover type as bypass
  - Application type HTTP
- Activates the services
- Activates the content rule

The script enables you to:

- Configure a service name (Transparent_Cache1)
- Define whether to direct only cacheable content or all content to the cache servers

To configure a transparent cache configuration, enter **4** at the Configuration Script menu.

```
Which feature do you want to configure?

[1] Layer3 load balancing
[2] Layer5 load balancing
[3] Proxy cache
[4] Transparent cache

Enter the number for the feature you want to configure: 4
```

To accept the script default values, press the **Enter** key at the prompts.

```
Creating Transparent Cache Configuration

Enter service name: [default = Transparent_Cache1]

Enter service IP address: [default = 0.0.0.0] 192.168.3.7
Create another service? [y/n]? n

Transparent caching can be configured to direct only cacheable content
to the cache server. Non-cacheable content is sent directly to the
origin server.

The alternative is to direct all traffic to the cache server
regardless of whether the content is cacheable.
Should only cacheable content be directed to the cache server? [y/n]?
```

Enter one of the following:

- **y** to define URL "/*" as eql-cacheable in the content rule and allow the CSS to direct only cacheable content to the cache servers.
- **n** to define URL "/*" in the content rule and allow the CSS to direct all content to the cache servers.

After you specify the configuration, the script automatically:

- Displays the running-config file
- Saves the running configuration to the startup-config file

```
Showing the Running Config

!Generated MAR 6 17:53:49

!**************** GLOBAL *****************
ip route 0.0.0.0 0.0.0.0 192.168.3.3
!**************** CIRCUIT ****************
circuit VLAN1
ip address 192.168.3.6 255.255.255.0
!**************** SERVICE ****************
service Transparent_Cache1
    ip address 192.168.3.7
    type transparent-cache
    port 80
    protocol tcp
    active
```

```
!****************** EQL ******************
eql Cacheable
    description "This EQL contains
        extensions of cacheable content"
    extension pdf "Acrobat"
    extension fdf "Acrobat Forms Document"
    extension au "Sound audio/basic"
    extension bmp "Bitmap Image"
    extension z "Compressed data
        application/x-compress"
    extension gif "GIF Image image/gif"
    extension html "Hypertext Markup
        Language text/html"
    extension htm
    extension js "Java script
        application/x-javascript"
    extension mocha
    extension jpeg "JPEG image image/jpeg"
    extension jpg
    extension jpe
    extension jfif
    extension pjpeg
    extension pjp
    extension mp2 "MPEG Audio audio/x-mpeg"
    extension mpa
    extension abs
    extension mpeg "MPEG Video video/mpeg"
    extension mpg
    extension mpe
    extension mpv
    extension vbs
    extension m1v
    extension pcx "PCX Image"
    extension txt "Plain text text/plain"
    extension text
    extension mov "QuickTime video/quicktime"
    extension tiff "TIFF Image image/tiff"
    extension tar "Unix Tape Archive
        application/x-tar"
    extension avi "Video for Windows
        video/x-msvideo"
    extension wav "Wave File audio/x-wav"
    extension gz "application/x-gzip"
    extension zip "ZIP file
        application/x-zip-compressed"
```

```
!***************** OWNER *****************
owner Transparent_Owner
content Transparent_Rule
    add service Transparent_Cache1
    port 80
    protocol tcp
    url "/*" eql Cacheable or url "/*"
    balance domain
    failover bypass
    application http
    active
##########################################
##    Setup Completed Successfully!!!   ##
##########################################
```

Rebooting the CSS

Use the **reboot** command to reboot the CSS. This command is supported in all modes except user mode.

Before you enter the **reboot** command, save an existing running-config file prior to rebooting the CSS by using the **copy running-config startup-config** command from SuperUser mode. If you are not in expert mode, the CSS displays the prompts to save profile and configuration changes before it reboots.To save an existing running-config file prior to rebooting the CSS, use the **copy running-config startup-config** command from SuperUser mode.

To reboot the CSS:

```
(config)# reboot
```

Shutting Down the CSS

Use the **shutdown** command to shut down the CSS. This command shuts down all CSS processes so you can power cycle the unit safely. The **shutdown** command is supported in all modes except in user mode.

To shut down the CSS:

```
(config)# shutdown
```

Where to Go Next

Chapter 2, Configuring CSS Basics describes the initial configuration procedures for the CSS, such as changing the administrative username and password, creating usernames and passwords, configuring the Ethernet management port, specifying a static IP address and subnet mask, and changing the date and time.

Configuring CSS Basics

This chapter describes the initial configuration procedures for the CSS. Information in this chapter applies to all CSS models, except where noted.

This chapter contains the following major sections:

- Initial Setup Quick Start
- Changing the Administrative Username and Password
- Creating Usernames and Passwords
- Configuring the Ethernet Management Port
- Configuring an IP Route
- Configuring Date, Time, and Time Zone
- Synchronizing the CSS with an SNTP Server
- Configuring a Host Name

Initial Setup Quick Start

Table 2-1 is a quick start configuration table designed to help you configure the CSS quickly and easily. This table provides the following basic steps:

- Log in and access config mode
- Change the default administrative username and password
- Create additional usernames and passwords to log in to the CSS (optional)
- Access boot mode to configure an IP address and subnet mask for the Ethernet management port
- Configure a static route for destination networks that are outside the local subnet of the CSS (optional)
- Configure a default IP route
- Enter the date, time, and time zone (optional)
- Specify a Simple Network Time Protocol (SNTP) server (optional)

Following Table 2-1 is an overview of the CSS system software and commands for the initial setup of the CSS.

Once you configure the Ethernet management port IP address, you can continue to use the console port or you can use the Ethernet management port to Telnet in to the CSS and configure it remotely.

Table 2-1 Initial Setup Quick Start

Task and Command Example
1. Log in to the CSS using the default administrative username **admin** and password **system**, or the username and password assigned to you during the boot process. Refer to Chapter 1, Booting, Logging In, and Getting Started for details on logging in to the CSS.

2. Access config mode.
`# config`
`(config)#`

3. Change the default administrative username and password. `(config)# username-offdm bobo password secret`

Table 2-1 Initial Setup Quick Start (continued)

Task and Command Example

4. Create usernames and passwords to log in to the CSS (optional). The CSS supports a maximum of 32 usernames, including the administrator and technician usernames. You can assign each user with SuperUser or User status.

```
(config)# username picard password “captain” superuser
```

5. Access boot mode to configure an IP address for the Ethernet management port. This IP address must be on a different subnet than any other CSS virtual LAN (VLAN) circuit IP subnet or you will not be able to access the port. You must reboot the CSS for the new IP address to take effect.

```
(config)# boot
(config-boot)# ip address 172.16.6.58
```

6. Configure a subnet mask for the Ethernet management port in boot mode.

```
(config-boot)# subnet mask 255.255.255.0
```

7. Exit from boot mode to config mode.

```
(config-boot)# exit
```

8. Configure a static IP route, as required.

```
(config)# ip route 192.168.3.123/16
```

9. Exit from config mode to configure a date. The **clock date** command does not allow backspacing. If you enter a wrong date, reenter the command with the new information.

   Enter the date in the format *mm-dd-yy.*

```
# clock date
Enter date: [12-31-03] 12-31-03
```

   To use the European format to specify the date (using the format of day, month, and year), access config mode and use the **date european-date** command to enable the **clock date** command to accept date input in the format of day, month, and year.

```
(config)# date european-date
(config)# exit
# clock date
Enter date: [31-12-03] 31/12/03
```

Table 2-1 Initial Setup Quick Start (continued)

Task and Command Example

10. Configure the time using the **clock time** command. The **clock time** command does not allow backspacing. If you enter the wrong time, reenter the command with the new information.

Enter the time in the format *hh*:*mm*:*ss*.

```
# clock time
Enter time: [15:17:33] 16:17:33
```

11. (Optional) Specify the time zone and Universal Time Coordinated (UTC) offset if you are using an SNTP server to synchronize the CSS system clock.

```
# clock timezone EST hours 3 before-UTC
```

12. (Optional) Access config mode and specify the SNTP server and the polling frequency if you are using an SNTP server to synchronize the CSS system clock.

```
# config
(config)# sntp server 192.168.19.21 version 2
(config)# sntp poll-interval 90
```

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

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

Changing the Administrative Username and Password

During the initial log in to the CSS you enter the default user name **admin** and the default password **system** in lowercase text. For security reasons, you should change the administrative username and password. Security on your CSS can be compromised because the administrative username and password are configured to be the same for every CSS shipped from Cisco Systems.

The administrative username and password are stored in nonvolatile random access memory (NVRAM). Each time you reboot the CSS, it reads the username and password from NVRAM and reinserts them in to the user database. SuperUser status is assigned to the administrative username by default.

You can change the administrative username and password, but because the information is stored in NVRAM, you cannot permanently delete them. If you delete the administrative username using the **no username** command, the CSS deletes the username from the running-config file, but restores the username from NVRAM when you reboot the CSS.

Use the **username-offdm** *name* **password** *text* command to change the administrative username or password.

Note You can also use the Security Options menu from the Offline DM menu (accessed during the boot process) to change the administrative username and password. Refer to Appendix B, Using the Offline Diagnostic Monitor Menu for information on the Offline DM menu.

For example, to change the default administrative username and password to a different username and password, enter.

```
(config)# username-offdm bobo password secret
```

Creating Usernames and Passwords

The CSS supports a maximum of 32 usernames, including an administrator username and a technician username. You can assign each user that logs into the CSS with SuperUser or User status.

- **User** - Allows access to a limited set of commands that enable you to monitor and display CSS parameters, but not change them. A User prompt ends with the **>** symbol. To view the commands available in User mode, at the User prompt, enter **?**.

  By default, new users have only user-level status unless you configure them to have SuperUser status.

- **SuperUser** - Allows access to the full set of CLI commands, including those in User mode, that enable you to configure the CSS. A SuperUser prompt ends with the **#** symbol.

  From SuperUser mode, you can enter global configuration mode and its subordinate configuration modes.

Use the **username** command to create usernames and passwords to log in to the CSS. The syntax for this global configuration mode command is:

```
username name [des-password|password] password {superuser}
    {dir-access access}
```

Note Any user with SuperUser status can create CSS usernames. To allow only administrator or technician users to create usernames, use the **restrict user-database** command (see Chapter 3, Managing the CSS Software).

The options and variables are as follows:

- *name* - Sets the username you want to assign or change. Enter an unquoted text string with no spaces and a maximum of 16 characters. To see a list of existing usernames, enter **username ?**.

- **des-password** - Specifies that the password you enter is the Data Encryption Standard (DES) form of the password. Use this option *only* when you are creating a script or a startup configuration file. Enter a DES-encrypted, case-sensitive, unquoted text string with no spaces from 6 to 64 characters.

Note If you specify the **des-password** option, you must know the encrypted form of the password to successfully log in to the CSS. You can find the CSS encrypted password in the Global section of the running-config. To display the running-config, use the **show running-config** command.

- **password** - Specifies that the password is not encrypted onyour display as you enter it. However, the CSS DES-encrypts the password in the running-config for extra security. Use this option when you use the CLI to create users. Enter a case-sensitive, unquoted text string with no spaces from 6 to 16 characters.
- *password* - The text string that you enter. The CSS allows all special characters in a password except for the percent sign (%).
- **superuser** - (Optional) Specifies SuperUser privileges to allow a user to access SuperUser mode. If you do not enter this option, the user can only access User mode.
- **dir-access** *access* - (Optional) Defines the CSS directory access privileges for the username. There are access privileges assigned to the seven CSS directories; Script, Log, Root (installed CSS software), Archive, Release Root (configuration files), Core, and MIBs. By default, users have both read- and write-access privileges (B) to all seven directories. Changing the access level also affects the use of the CLI commands associated with directories.

  Enter one of the following access privilege codes for the CSS Script, Log, Root, Archive, Release Root, Core, and MIB directories, in this order:

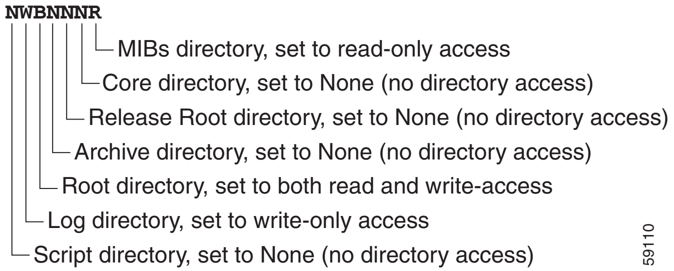
  - **R** - Read-only access to the CSS directory
  - **W** - Write-only access to the CSS directory
  - **B** - Both read- and write-access privileges to the CSS directory
  - **N** - No access privileges to the CSS directory

The following example creates a SuperUser named *picard* with a password of *captain*.

```
(config)# username picard password “captain” superuser
```

Figure 2-1 shows how the access privilege settings corresponds to the CSS directories.

Figure 2-1 CSS Directory Access Privileges


NWBNNNR
MIBs directory, set to read-only access
Core directory, set to None (no directory access)
Release Root directory, set to None (no directory access)
Archive directory, set to None (no directory access)
Root directory, set to both read and write-access
Log directory, set to write-only access
Script directory, set to None (no directory access)
59110

For example, to define directory access for username *picard*, enter:

```
(config)# username picard password "captain" superuser NWBNNNR
```

To display a list of existing usernames, enter:

```
(config)# username ?
```

To remove an existing username, enter:

```
(config)# no username picard
```

To change a user password, reenter the **username** command and specify the new password. Remember to include SuperUser privileges if required. For example:

```
(config)# username picard password "flute" superuser
```

Caution The **no username** command removes a user permanently. Make sure you want to perform this action because you cannot undo this command.

Configuring the Ethernet Management Port

You can directly communicate with the CSS and enter command line interface (CLI) commands using the Ethernet management port. You must assign an IP address and a subnet mask to be able to access the Ethernet management port. You can also configure an Ethernet management port default gateway to load a boot file on a CSS across different subnets.

The Ethernet management port is located on the:

- CSS 11501 front panel
- CSS 11503 and CSS 11506 SCM front panels

The CSS enables you to configure an IP address, a subnet mask, and a default gateway:

- At the prompts during the boot process
- Using the Offline DM menu
- Using CLI commands

For information on configuring an IP address, subnet mask, and default gateway for the Ethernet management port using CLI commands, see the following sections. Refer to Appendix B, Using the Offline Diagnostic Monitor Menu for information on using the Offline DM Main menu to configure an IP address, subnet mask, and default gateway during the boot routine.

Note Access control lists (ACLs) are not supported on the CSS Ethernet management port.

This section includes the following topics:

- Configuring an IP Address for the Ethernet Management Port
- Configuring a Subnet Mask
- Configuring a Management Port Default Gateway

Configuring an IP Address for the Ethernet Management Port

Use the **ip address** command to configure an IP address for the CSS Ethernet management port. This command is available in boot mode. The **ip address** command does not have a **no** version. To change the IP address, reenter the **ip address** command and enter the new IP address.

You must reboot the CSS for the new IP address to take effect.

Note The Ethernet management port IP address must be on a different subnet than any other CSS VLAN circuit IP subnet. If you do not make the Ethernet management port IP address unique, you will not be able to access the port.

An IP address of 0.0.0.0 for the Ethernet management port is a legal setting and disables the management port upon reboot. If you enter 0.0.0.0, and attempt to use the **subnet mask** command, the following message appears:

```
The mask cannot be set because the IP address is 0.0.0.0.
```

For example, to specify an Ethernet management port IP address in boot mode, enter:

```
    (config)# boot
    (config-boot)# ip address 172.16.6.58
```

Configuring a Subnet Mask

Use the **subnet mask** command to configure the CSS subnet mask for the Ethernet management port. This command is available in boot mode.

You must reboot the CSS for the new subnet mask to take effect.

For example, to specify an Ethernet management port subnet mask of 255.255.255.0, enter:

```
    (config)# boot
    (config-boot)# subnet mask 255.255.255.0
```

To remove the configured subnet mask, enter:

```
    (config-boot)# no subnet mask
```

Configuring a Management Port Default Gateway

Use the **gateway address** command to configure an Ethernet management port default gateway for use in Offline DM. The **gateway address** command allows you to boot the CSS from the Offline DM when the boot image resides on a different subnet. This command is available in boot mode.

To disable the default gateway address and set it to an IP address of 0.0.0.0, use the **no** form of the **gateway address** command. A gateway address of 0.0.0.0 for the Ethernet management port does not appear in the **show boot-config** command output for the CSS boot configuration.

To specify a default gateway address for the Ethernet management port for use in Offline DM, enter:

```
(config)# boot
(config-boot)# gateway address 172.16.57.2
```

To disable the default gateway address and set the IP address to 0.0.0.0, enter:

```
config-boot)# no gateway address
```

If you have a second SCM installed in a CSS 11503 or CSS 11506, use the **passive gateway address** command to configure the management port gateway address in the passive SCM boot-config (see Chapter 4, Specifying the CSS Boot Configuration).

Configuring an IP Route

To establish IP connectivity to the CSS, a static IP route is required to connect the CSS to next hop router. A static route consists of a destination network address and mask and the next hop to reach the destination. You can also specify a default static route (using 0.0.0.0 as the destination network address and a valid next hop address) to direct frames for which no other destination is listed in the routing table. Default static routes are useful for forwarding otherwise unrouteable packets by the CSS.

When you configure a static IP route, the CSS periodically polls the next hop router with an internal ICMP keepalive service to ensure the router is functioning properly. If the router fails, the CSS removes any entries from the routing table that point to the failed router and stops sending traffic to the failed router. When the router recovers, the CSS:

- Becomes aware of the router
- Reenters applicable routes in to the routing table

To configure a static IP route, use the **ip route** command and specify one of the following:

- An IP address and prefix length; for example, 192.168.1.0 /24
- An IP address and a subnet mask; for example, 192.168.1.0 255.255.255.0

The syntax for the **ip route** command is:

ip route *ip_address subnet mask ip_address2*

The variables are as follows:

- *ip_address* - The destination network address. Enter the IP address in dotted-decimal notation (for example, 192.168.11.1).
- *subnet_mask* - The IP subnet mask. Enter the mask as either:
  - A prefix length in CIDR bit-count notation (for example, /24)
  - An IP address in dotted-decimal notation (for example, 255.255.255.0)
- *ip_address2* - The next hop address for the route. Enter the IP address in dotted-decimal notation (for example, 192.168.11.1).

For example, to configure a static IP route to destination network address *192.168.0.0 /16* and a next hop address of *192.168.1.1*, enter:

```
(config)# ip route 192.168.0.0 /16 192.168.1.1
```

For example, to configure a default IP route using a destination address of *0.0.0.0/0* and a next hop address of *192.167.1.1*, enter:

```
(config)# ip route 0.0.0.0 /0 192.167.1.1
```

Refer to Chapter 6, Configuring CSS Network Protocols for complete information on configuring IP routes.

Configuring Date, Time, and Time Zone

Use the **clock** command to set the date, time, or time zone for the CSS. When you enter this command, the CSS displays the current date and time.

The **clock** command does not allow backspacing. If you enter the wrong date, time, or time zone, you must reenter the command with the new information.

This section includes the following topics:

- Setting the Date
- Setting the European Date
- Setting the Time
- Setting the Time Zone
- Showing the Date and Time

Setting the Date

Use the **clock date** command to set the date. A prompt appears to show the current date in the correct format to use. Enter the month, day, and year as integers with dash characters separating them. For example, enter June 15th 2003 as 06-15-03.

Enter the new date in the format *mm-dd-yy* as shown:

```
# clock date
Enter date: [12-31-03] 12-31-03
```

Setting the European Date

Use the **date european-date** global configuration mode command to specify the date in the European format of day, month, and year. This command enables the **clock date** command to accept the date in day, month, and year, separated by slashes (/).

Enter the new date in the format *dd/mm/yy* as shown:

```
(config)# date european-date
(config)# exit
# clock date
Enter date: [31-12-03] 31/12/03
```

To reset the format for the **clock date** command to the default of month, day, and year, enter:

```
(config)# no date european-date
```

Setting the Time

Use the **clock time** command to set the time. This command sets the time in military-time (24-hour) format. A prompt appears to show the current time in the correct format to use. Enter the hour, minutes, and seconds as integers, separated by colons.

Enter the new time in the format *hh*:*mm*:*ss* as shown:

```
# clock time
Enter time: [15:12:38] 16:12:38
```

Setting the Time Zone

Use the **clock timezone** command to specify a time zone for the CSS, which synchronizes the CSS system clock with an SNTP server. The time stored in the CSS is the local time. The SNTP server calculates the Universal Time Coordinated (UTC, also known as Greenwich Mean Time) time by offsetting the time zone from the local time. If required, you can apply a negative offset to the UTC (for example, –05:-23:+00) or a positive offset to the UTC (for example, +12:+00:+00).

Use the **no** form of the **clock timezone** command to reset the time zone information to 00:00:00, and also to set the clock to the new time without the time zone offset.

Note The use of the **clock timezone** command assumes you are using the CSS with an SNTP server to synchronize the CSS system UTC time to that of a designated SNTP server. Without a configured SNTP server, the time zone information is not used. See the "Synchronizing the CSS with an SNTP Server" section for details.

The syntax for the **clock timezone** command is:

clock timezone *name* **hours** *hours* {**before-UTC|after-UTC**} {**minute** *minutes* {**before-UTC|after-UTC**}

The options and variables are as follows:

- **timezone** *name* - The name of the time zone. Enter a name with a maximum of 32 characters and no spaces.
- **hours** *hours* - The hours of offset for the time zone. Enter a number from 0 to 12. Use with the **before-UTC** option or **after-UTC** option to set the offset to either a negative or positive number.
- **before-UTC -** The offset for UTC as a negative number. For example, if the hour offset is 12, **before-UTC** sets the offset to –12.
- **after-UTC** - The offset for UTC as a positive number (the default offset).
- **minute** *minutes* - The minutes of offset for the time zone. Enter a number from 0 to 59. Use with the **before-UTC** option or **after-UTC** option to set the offset to either a positive or negative number.

For example, to enter the new time zone for Eastern Standard Time (EST) with a –3 hour offset:

```
# clock timezone EST hours 3 before-UTC
```

To set the time zone offset back to 00:00:00 (and also set the clock to the new time without the time zone offset):

```
# no clock timezone
```

Showing the Date and Time

Use the **show clock** command to display the current date and time. For example:

```
# show clock
```

Table 2-2 describes the fields in the **show clock** command output.

Table 2-2 Field Descriptions for the show clock Command

Field	Description
Date	The configured date in the format of month, day, and year; for example, the date June 15th 2003 appears as 06-15-2003. If you use the **date european-date** command, the format is day, month, and year. For example, the date June 15th 2003 appears as 15-06-2003.
Time	The configured time in the format of hour, minute, and second; for example, 16:23:45. If you configure an SNTP server, the **show clock** command displays the time adjusted with the time zone offset. The **show clock** command displays the UTC time from the SNTP server. If you configure a time zone, the **show clock** command displays the time adjusted with the time zone offset. For example, if the UTC time from the server is 16:30:43 and you configure a time zone negative offset of 5 hours and 30 minutes (–05:-30:+00), the displayed time becomes 11:00:43.
Timezone	The configured time zone offset from an SNTP server. All zeros (00:00:00) indicate that no offset was configured for the time zone. A negative symbol (–) indicates a negative offset to the UTC (for example, -05:-23:+00). A positive symbol (+) indicates a positive offset to the UTC (for example, +12:+00:+00).

Synchronizing the CSS with an SNTP Server

Use the **sntp** command to configure the Simple Network Time Protocol (SNTP) on the CSS. Use SNTP when you need to synchronize computer system clocks on the Internet to that of a designated SNTP server. SNTP is a simplified, client-only version of the Network Time Protocol (NTP) that enables the CSS time-of-day to be synchronized with any SNTP server.

Accurate time-of-day is provided by synchronizing to the UTC time, which provides time within 100 milliseconds of the accurate time. You can configure information about the local time zone so the time appears correctly relative to the local time zone. The CSS can receive the time from only a single SNTP server (in unicast mode), but the CSS cannot be used to provide time services to other devices.

Before you synchronize the CSS with an SNTP server, make sure you configure the proper time zone for the CSS (for example, to EST). Also make sure the time difference between the CSS internal clock and the SNTP server clock is less than 24 hours. Otherwise, the CSS will not synchronize its clock with the SNTP server. To configure the time on the CSS, see the "Configuring Date, Time, and Time Zone" section for details.

For detailed information on configuring the SNTP server, consult the documentation provided with the server.

This section includes the following topics:

- Configuring the SNTP Server
- Configuring the SNTP Poll Interval
- Showing SNTP Configuration Information

Configuring the SNTP Server

Use the **sntp server** command to specify the SNTP server. The syntax for this command is:

sntp server *ip_address* **{version** *number***}**

The options and variables are as follows:

- **server** *ip_address* - The IP address for the SNTP server. Enter an IP address in dotted-decimal notation (for example, 192.168.1.0).
- **version** *number* - The version number of the SNTP server. Enter a version number between 1 and 4. The default is 1.

To configure an SNTP server (running version number 3), enter:

```
(config)# sntp server 192.168.19.21 version 3
```

To remove the specified SNTP server, enter:

```
(config)# no sntp server
```

Configuring the SNTP Poll Interval

Use the **sntp poll-interval** command to specify the poll interval for SNTP request messages. The poll interval is the time (in seconds) between successive SNTP request messages to the server. Continuous polling is critical for the CSS to obtain time from the SNTP server and ensure the local time matches the “real time” of the server. The valid entries are 16 to 16284 seconds. The default is 64 seconds.

To specify an SNTP poll-interval of 90 seconds, enter:

```
(config)# sntp poll-interval 90
```

To return the SNTP poll-interval to the default setting of 64 seconds, enter:

```
(config)# no sntp poll-interval
```

Showing SNTP Configuration Information

To display the SNTP configuration information on the CSS, enter the **show sntp global** command. For example:

```
(config)# show sntp global
```

Table 2-3 describes the fields in the **show sntp global** command output.

Table 2-3 Field Descriptions for the show sntp global Command

Field	Description
Server Address	The IP address for the SNTP server.
Version	The version number of the server. The default is 1.
Poll Interval	The time in seconds between SNTP request messages. The range is 16 to 16284. The default is 64.
TimeSinceLastUpdate	The time in seconds since the last server reply.
Server Status	The operating status of the SNTP server, Up or Down.

Configuring a Host Name

Use the **host** command to manage entries in the Host table. The Host table is the static mapping of mnemonic host names to IP addresses, which is analogous to the ARP table.

The syntax for this global configuration mode command is:

host *host_name ip_address*

The variables are as follows:

- *host_name* - The name of the host. Enter an unquoted text string with no spaces and a length from 1 to 16 characters.
- *ip_address* - The address associated with the host name. Enter the IP address in dotted-decimal notation (for example, 192.168.11.1).

To add a host to the Host table, the host name must not exist in the Host table. To change a current host address, remove the host name and then add it again to the Host table.

For example:

```
(config)# host CSS11501-LML 192.168.3.6
```

To remove the existing host from the Host table, enter:

```
(config)# no host CSS11501-LML
```

To display a list of host names, enter:

```
(config)# show running-config global
```

Where to Go Next

Chapter 3, Managing the CSS Software provides details on managing the CSS software. It discusses the use of the running-config and startup-config files, specifying file storage locations for a two-disk CSS, unpacking and removing an ArrowPoint Distribution Image (ADI), and archiving files. Also included in this chapter is an overview of the CSS system software.

Managing the CSS Software

This chapter describes how to manage the software running on the CSS. Information in this chapter applies to all CSS models, except where noted.

This chapter contains the following major sections:

- CSS Software Overview
- Restricting Modifications to the CSS User Database
- Creating an FTP Record
- Using the Running-Config and Startup-Config Files
- Configuring Disks in a Two-Disk CSS
- Unpacking and Removing an ADI
- Archiving Files to the Archive Directory
- Restoring Files from the Archive Directory
- Enabling and Copying Core Dumps
- Showing CSS Configurations

CSS Software Overview

The CSS software contains the files needed to run the CSS, including boot files, directories for archiving and logging files, and MIB files. This software is pre-installed on the CSS conventional hard disk or on an optional Flash disk, which is a Flash memory-based storage device. The CSS software is approximately 50 MB, and you can install a maximum of two software versions.

The CSS software image is available from the Cisco Systems Web site (www.cisco.com) as an ArrowPoint Distribution Image (ADI), network boot ZIP (.zip) image, or GZIP-compressed (adi-gz) image.

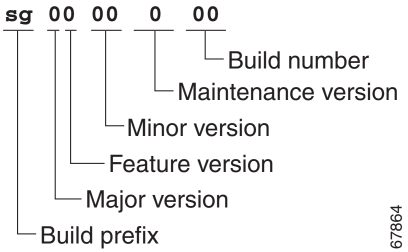
You can install the CSS software on an FTP server, which the CSS accesses through the File Transfer Protocol (FTP). The CSS accesses the ADI or GZIP file containing the CSS software from an FTP server, copies the file to the CSS disk, and unpacks it. The CSS then boots from the disk.

You can also install the CSS software on a network-mounted drive on a remote system, which the CSS accesses through FTP. Network boot uses a special ZIP version of WebNS that ends with a .zip extension. Instead of the CSS disk, the network file system contains the CSS software. This software must be copied and uncompressed on the network drive.

Refer to Chapter 4, Specifying the CSS Boot Configuration for information on booting the CSS, including from a network boot drive.

The CSS software version format is defined as shown in Figure 3-1.

Figure 3-1 Software Version for the CSS



To display the software versions installed on the CSS, use the **show version** and **show installed-software** commands, as described in the "Showing Software Information" section.

From an FTP server, you can view the following directories on the hard disk or Flash disk:

- The log directory contains the following log files:
  - **boot.log** - ASCII log of the boot process
  - **boot.bak** - Backup of the previous boot log
  - **sys.log** - ASCII log of system events (logging to disk is enabled by default to **subsystem all** and **level info**)
  - **sys.log.prev** - Backup of the previous system log file (if any)
- The scripts directory contains default, profile, and sample scripts.
- The core directory contains any core dumps created by the CSS. For information on copying core dumps to an FTP or TFTP server, see the "Enabling and Copying Core Dumps" section.
- The MIB directory contains MIB files that you can load in to SNTP-compliant network management software applications.

⚠ **Caution** When you view the CSS software directories installed on a network drive, more directories are listed than those you can view on the hard disk or Flash disk. The additional directories are reserved for internal use. Do not manipulate the files in these directories.

The software directory also contains the startup-config file. The startup-config is an ASCII file containing commands that the CSS executes at startup. This file is created when you:

- Finish using the Configuration Script (refer to Chapter 1, Booting, Logging In, and Getting Started).
- Use the **copy running-config startup-config** or **write memory** command (see the "Saving the Running-Config to the Startup-Config File" section). Both commands save configuration changes to the startup-config file during a CSS session. The **write memory** command also archives the startup configuration file to the archive directory on the CSS (similar to the **archive startup-config** command, see the "Archiving Files to the Archive Directory" section).
- Use FTP to copy a startup-config file to the CSS.

The archive directory contains the files that you archive from the current software by using the **archive** command. These files include the running-config file, startup-config file, log files, profile scripts, and scripts you create. You can view a list of archived files by using the **show archive ?** command.

To restore any archived files to the CSS, use the **restore** command. For more information on the **archive** and **restore** commands, see the "Archiving Files to the Archive Directory" and "Restoring Files from the Archive Directory" sections.

Restricting Modifications to the CSS User Database

By default, access to the CSS user database is not restricted. Nonrestricted access means any user with SuperUser privileges (local user, administrator, or technician) can:

- Create, modify, or delete usernames (user database entries)
- Clear the CSS running-config file

You can use the **restrict user-database** command to restrict the CSS user database to CSS users who are identified as either an administrator or a technician.

To restrict modification of the CSS user database, enter:

```
(config)# restrict user-database
```

To remove restrictions for modifying the CSS user database, enter:

```
(config)# no restrict user-database
```

Creating an FTP Record

Use the **ftp-record** command to create a File Transfer Protocol (FTP) record file to use when accessing an FTP server from the CSS. The syntax for this global configuration mode command is:

> **ftp-record** *ftp_record ip_address_or_hostname username*
> ["*password*"|**des-password** ***des_password***|**encrypted-password**
> *encrypted_password*] {*base_directory*}

Note The CSS FTP server supports only the active (normal) FTP mode of operation. It does not support the passive FTP mode of operation.

The variables for this command are as follows:

- *ftp_record* - The name for the FTP record file. Enter an unquoted text string with no spaces and a maximum of 16 characters.
- *ip_address_or_hostname* - The IP address or host name of the FTP server you want to access. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1) or a mnemonic host name (for example, myhost.mydomain.com).
- *username* - A valid login username on the FTP server. Enter a case-sensitive unquoted text string with no spaces and a maximum of 16 characters.
- *password* - The password for the valid login username on the FTP server. Enter a case-sensitive quoted text string with no spaces and a maximum of 16 characters.
- *des_password* - The Data Encryption Standard (DES) encrypted password for the valid login username on the FTP server. Enter a case-sensitive unquoted text string with no spaces and a maximum of 64 characters.
- *encrypted_password* - The encrypted password for the valid login username on the FTP server. Enter a case-sensitive unquoted text string with no spaces and a maximum of 16 characters.
- *base_directory* - An optional base directory for this record. Enter the base directory name as a case-sensitive unquoted text string with no spaces and a maximum of 64 characters.

The config-path and base directory path in the FTP record associated with a network boot must not contain a pathname that conflicts with a non-network drive name (for example, c: or host:).

For example (using an encrypted password), to create an FTP record called *arrowrecord*, enter:

```
# ftp-record arrowrecord 192.168.19.21 bobo password “secret”
/outgoing
```

To delete the FTP record *arrowrecord* from the CSS, enter:

```
# no ftp-record arrowrecord
```

Copying Files from an FTP Server

Use the **copy ftp** command to copy files from an FTP server to the CSS. This command is available in SuperUser mode. Before using this command, you must use the **ftp-record** global configuration mode command to create an FTP record file containing the FTP server IP address, username, and password.

The syntax for this command is:

copy ftp *ftp_record filename* [**boot-image**|**script***script_filename*|
startup-config]

The options and variables for this command are as follows:

- *ftp_record* - Name of the FTP record file that contains the FTP server IP address, username, and password. Enter an unquoted text string with no spaces. To create an FTP record, use the **ftp-record** global configuration mode command.
- *filename* - Name of the file on the FTP server that you want to copy to the CSS. Include the full path to the file. Enter an unquoted text string with no spaces and a maximum length of 32 characters.

  If you are using the **boot-image** keyword to copy an ADI file from an FTP server to the CSS, include the full path to the file including the file extension. Enter an unquoted text string with no spaces and a maximum length of 32 characters. You can also copy a GZIP-compressed version of the ADI file. The CSS uncompresses the file. If there is not enough disk space available, the CSS provides a message.

- **boot-image** - Copies an ADI file from an FTP server. The ADI file contains the CSS software including boot files and logging and archiving directories. To unpack the CSS software in the ADI file, use the **unpack** boot mode command. When you use the **boot-image** keyword, the file you copy to the CSS must be an ADI file. Otherwise, the CSS rejects it.
- **script** *script_file* - Copies an FTP file to the script directory. To assign a name to the script file on the CSS, enter an unquoted text string with no spaces and a maximum length of 32 characters.
- **startup-config** - Copies the startup-config file and overwrites the existing configuration file.

Using the Running-Config and Startup-Config Files

When you make configuration changes, the CSS places those changes in a virtual running configuration file (running-config). Before you log out or reboot the CSS, you must copy the contents of the running-config file to the startup-config file (startup-config) to save configuration changes. The CSS uses the startup configuration file on subsequent reboots.

This section includes the following topics:

- Saving the Running-Config to the Startup-Config File
- Copying the Running- and Startup-Config Files
- Clearing the Running-Config and Startup-Config Files
- Showing the Running Configuration
- Showing the Startup Configuration
- Creating a Running-Config or Startup-Config File Using a Text Editor
- Finding an IP Address in the Running-Config File

Saving the Running-Config to the Startup-Config File

To save the running-config file to the startup-config file on the CSS disk, use one of the following commands:

- **copy running-config startup-config** - Copies the contents of the running-config file to the startup-config file. The CSS uses the startup configuration upon reboot. If you do not copy the contents of the running-config file to the startup-config file before you reboot, changes to the running configuration are lost. This command is available in SuperUser mode.
- **write memory** - Copies the contents of the running-config file to the startup-config file (similar to the **copy running-config startup-config** command). In addition, the **write memory** command also archives the startup configuration file to the archive directory on the CSS (similar to the **archive startup-config** command, see the "Archiving Files to the Archive Directory" section).
- **copy startup-config running-config** - Copies the contents of the startup-config file to the running-config file and merges the contents with the running-config file. This command is available in SuperUser mode.

Copying the Running- and Startup-Config Files

The **copy running-config** command can also copy the running configuration to an FTP or TFTP server. This command is available in SuperUser mode.

Note If desired, use the **save_config** alias command to automatically copy the contents of the running-config file to the startup-config file, and then archive the startup-config file to the CSS disk.

The syntax for this command is:

copy running-config [[**ftp** *ftp_record*|**tftp** *ip_or_host*]*filename*|
startup-config]

The options and variables for this command are as follows:

- **ftp** *ftp_record filename* - Copies the running-config file to an FTP server. The name of the FTP record file contains the FTP server IP address, username, and password. Enter an unquoted text string with no spaces. To create an FTP record, use the **ftp-record** global configuration mode command.
- **tftp** *ip_or_host* - Copies the running-config file to a TFTP server. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1) or in mnemonic host-name format (for example, myhost.mydomain.com).
- *filename* - Name you want to assign to the file on the server. Include the full path to the file. Enter an unquoted text string with no spaces and a maximum length of 32 characters.
- **startup-config** - Copies the running-config file to the startup-config file on the CSS disk. In the event of the CSS rebooting, if you do not save changes in the running-config file to the startup-config file, these changes are lost.

The **copy startup-config** command can copy the startup configuration to an FTP or TFTP server. This command is available in SuperUser mode.

The syntax for this command is:

copy startup-config [[**ftp** *ftp_record*|**tftp** *ip_or_host*]*filename*|
running-config]

The options and variable for this command are as follows:

- **ftp** *ftp_record* - Copies the startu- configuration file to an FTP server. The name of the FTP record file contains the FTP server IP address, username, and password. Enter an unquoted text string with no spaces. To create an FTP record, use the **ftp-record** global configuration mode command.
- **tftp** *ip_or_host* - Copies the startup-config file to a TFTP server. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1) or in mnemonic host-name format (for example, myhost.mydomain.com).
- *filename* - Name you want to assign to the file on the server. Include the full path to the file. Enter an unquoted text string with no spaces and a maximum length of 32 characters.
- **running-config** - Copies the startup configuration and merges with the running configuration file on the CSS disk.

Clearing the Running-Config and Startup-Config Files

To reset the CSS running configuration to the default configuration, use the **clear running-config** command in SuperUser mode. This command takes effect immediately. The **clear running-config** command resets all configurations to their defaults.

Use of the **clear running-config** command is restricted to CSS users who are identified as either administrators or technicians.

For example:

```
# clear running-config
```

To reset the startup configuration to the default configuration, use the **clear startup-config** command in SuperUser mode. This command takes effect upon the next reboot. For example:

```
# clear startup-config
```

Showing the Running Configuration

To display the CSS running configuration, use the **show running-config** command. Configuration entries within each mode in the running-config file (such as Global, Interface, Circuit, and Service) appear in chronological order, based on the order in which you configure the CSS. The CSS does not display default configurations in the CSS running configuration.

The syntax and options for the **show running-config** command are as follows:

- **show running-config** - Displays all components of the running-config file.
- **show running-config acl** {*index number*} - Displays access control list (ACL) information in the running-config file. For information about a specific ACL, include its index number.
- **show running-config circuit** {*circuit name*} - Displays the circuit components of one or all circuits in the running-config file.
- **show running-config dql** {*dql name*} - Displays domain qualifier list (DQL) information of the running-config file. For information about a specific DQL, enter the DQL name as a case-sensitive unquoted text string.

- **show running-config eql** {*eql name*} - Displays extension qualifier list (EQL) information of the running-config file. For information about a specific EQL, enter the EQL name as a case-sensitive unquoted text string.
- **show running-config global** - Displays the global components of the running-config file.
- **show running-config group** {*group name*} - Displays the valid existing group components of the running-config file. For information about a specific group, enter the group name as a case-sensitive unquoted text string.
- **show running-config header-field-group** {*name*} - Displays the valid existing header-field group components of the running-config file. For information about a specific group, enter *name* as a case-sensitive unquoted text string with a maximum of 16 characters. To see a list of header-field groups, enter **show running-config header-field-group ?**.
- **show running-config interface** *interface name* - Displays a specific interface component of the running-config file.
  - For a CSS 11501, enter the interface name in interface-port format (for example, e2)
  - For a CSS 11503 or CSS 11506, enter the interface name in slot/port format (for example, 3/1)
- **show running-config interfaces** - Displays all the interface components of the running-config file.
- **show running-config keepalive** {*keepalive name*} - Displays the existing keepalive components of the running-config file. For information about a specific keepalive, enter *keepalive_name* as a case-sensitive unquoted text string and a maximum of 32 characters. To see a list of keepalives, enter **show keepalive-summary**.
- **show running-config nql** {*name*} - Displays network qualifier list (NQL) information of the running-config file. For information about a specific NQL, enter the NQL name as a case-sensitive unquoted text string.
- **show running-config owner** {*owner name*} - Displays the valid existing owner components of the running-config file. For information about a specific owner, enter the owner name as a case-sensitive unquoted text string.
- **show running-config rmon-alarm** - Displays RMON alarm information of the running-config file.

- **show running-config rmon-event** - Displays RMON event information of the running-config file.
- **show running-config rmon-history** - Displays RMON history information of the running-config file.
- **show running-config service** {*service name*} - Displays the components of the running-config file for a valid existing service. For information about a specific service, enter the service name as a case-sensitive unquoted text string.
- **show running-config ssl-proxy-list** {*list_name*} - Displays RMON history information of the running-config file. Displays the components of the running configuration for a valid existing SSL-proxy list. For information about a specific list, enter *list_name* as a case-sensitive unquoted text string.
- **show running-config urql** {*urql name*} - Displays the components of the running-config file for existing uniform resource locator qualifier lists (URQL). For information about a specific URQL, enter the URQL name as a case-sensitive unquoted text string.

The following example shows a running-config file. Comments are preceded by an exclamation point (!). Note that the CSS does not display default values in the CSS running configuration or startup configuration even if you manually enter the values.

```
# show running-config
!************************ GLOBAL **********************
ip route 0.0.0.0/0 158.3.7.2
!********************** INTERFACE *********************
interface e1
   bridge vlan 2
interface e2
   bridge vlan 2
!*********************** CIRCUIT **********************
circuit VLAN1
   ip address 10.3.6.58 255.255.255.0
circuit VLAN2
   ip address 158.3.7.58 255.255.255.0
!*********************** SERVICE **********************
service serv1
   ip address 10.3.6.1
   active
service serv2
   ip address 10.3.6.2
   active
!************************ OWNER ***********************
owner arrowpoint.com
   content rule1
    ip address 158.3.7.43
    protocol tcp
    port 80
    add service Serv1
    add service Serv2
    active
```

Showing the Startup Configuration

Once you copy the contents of the running-config file to the startup-config file, use the **show startup-config** command to display the CSS startup configuration. The CSS does not display default configurations in the startup-config file.

Use the **show startup-config line-numbers** command to display the startup-config file with line numbers

The following example shows a CSS startup configuration with line numbers. Comments are preceded by an exclamation point (!).

```
# show startup-config line-numbers

1. !Generated MAR 6 18:56:11
2. configure
3. !********************** CIRCUIT **********************
4. circuit VLAN1
5.  ip address 192.168.2.170 255.255.255.0
6.  ip address 192.168.1.108 255.255.255.0
7. !********************** SERVICE **********************
8. service s1
9.  ip address 192.168.2.4
10. keepalive type none
11. active
12. !*********************** OWNER ***********************
13. owner rose
14. content rule-L3
15.  vip address 192.168.128.108
16.  add service s1
17.  active
18.  content rule-L5
19.  add service s1
20.  vip address 192.168.128.108
21.  url "/*"
22.  active
```

Creating a Running-Config or Startup-Config File Using a Text Editor

If you create a running- or startup-config file using a text editor, you must arrange the configuration information in the same order as occurs in an automatically created running- or startup-config file. The CSS arranges configuration information in the following categories within the running-config file and the startup-config file:

- Global - Configuration information relating to the CSS (for example, default route IP address)
- Interface - Physical port and VLAN associations
- Circuit - Circuit VLAN IP addresses and subnet masks
- SSL Proxy List - The ssl-proxy-list configuration
- Keepalive - The global keepalive configuration
- Service - Service names, IP addresses, and all service configuration information
- EQL - Extension Qualifier List (EQL) configuration
- Owner - Owner name, content rule name, and content rules
- Group - Source group configurations
- RMON Event - RMON event configurations
- RMON Alarm - RMON alarm configurations
- RMON History - RMON history configurations
- ACL - Access Control List (ACL) configurations
- URQL - Uniform Resource Locator Qualifier List (URQL) configurations

Though the CSS automatically organizes configuration information, the order in which you configure the CSS is important because of interdependencies within CSS functionality. Enter configuration commands for features in the same sequence as they appear in the startup-config file.

Finding an IP Address in the Running-Config File

Use the **find ip address** command to search the CSS running-config file for a specified IP address. You can include a netmask for subnet (wildcard) searches. This search can help you avoid IP address conflicts when you configure the CSS.

When you use this command, the CSS checks all services, source groups, content rules, ACLs, the management port, syslog, Application Peering Protocol (APP) sessions, and local interfaces in the running-config file for the specified IP address. If the address is found, the CSS displays the locations of its use. If no addresses are found, the CSS returns you to the command prompt.

This command is available in all modes. The syntax is:

find ip address *ip_or_host* {*subnet_mask*|**range** *number*}

The options and variables for this command are as follows:

- *ip_or_host* - IP address in dotted-decimal notation (for example, 192.168.11.1) or enter the host name in mnemonic host-name format (for example, host.domain.com).
- *subnet mask* - The IP subnet mask. Enter the subnet mask as either:
  - A prefix length in CIDR bit-count notation (for example, /24). Enter a prefix length of /16 or greater. Do not include a space to separate the IP address from the prefix length.
  - An IP address in dotted-decimal notation (for example, 255.255.255.0).
- **range** *number* - Defines how many IP addresses you want to find, starting with the *ip_or_host* address. Enter a number from 1 to 65535. The default range is 1.

  For example, if you enter an IP address of 192.168.1.1 with a range of 10, the CSS tries to find the addresses from 192.168.1.1 through 192.168.1.10.

For example:

```
(config)# find ip address 192.168.0.0

Users of IP address 192.168.0.0
Content Rule - 192.168.12.1, layer 3, owner: lml, state:Active
Content Rule - 192.168.12.1, layer 5, owner: lml, state:Active
Service - 192.168.3.6, serv1, state:Active
Service - 192.168.3.7, serv3, state:Active
Interface - 192.168.1.117. VLAN1
Interface - 192.168.2.117. VLAN1
```

Configuring Disks in a Two-Disk CSS

The CSS 11501 and the Switch Control Module (SCM) in the CSS 11503 and CSS 11506 contain two PCMCIA slots for a hard disk or Flash disk. These disks contain the CSS system software and are used for logging and storing offline system files. The two disks are identified by the PCMCIA slots (slot 0 and slot 1) in which they are installed. Disk 0 is the default storage location for the primary and secondary boot records in the CSS. The default storage location for log files and core dumps in the CSS is the specified disk from which the CSS boots (disk 0 or disk 1).

In addition to specifying the file storage locations, you can also:

- Format the disks
- Copy information such as the scripts, archives, or startup configuration from one disk to the other disk
- Display the mapping configuration of the two disks in slot 0 and slot 1
- Display the specified archive, log, script, or startup configuration file stored on a specific disk
- Delete a specific file (startup configuration, logs, scripts, or archive file) stored on a specific disk

This section includes the following topics:

- Formatting a Disk
- Specifying a Disk for Booting, Logging, and Core Dumps
- Copying Files Between Disks
- Showing the Disk Mapping Configurations
- Showing Files from a Disk
- Clearing Files from a Disk

As an alternate procedure for configuring disks from the CLI, you can use the Advanced Options menu of the Offline DM menu to reformat or set the disk mapping for the disks in slots 0 and 1. Refer to Appendix B, Using the Offline Diagnostic Monitor Menu for details.

Formatting a Disk

Use the **format** command to format and create the Core and Archive directories on a specified disk. The **format** command permanently erases all data on the disk. This command is available only in SuperUser mode.

If you wish to retain the startup-config file, ensure you move the file off the CSS before reformatting the disk. Also make sure you have a copy of the CSS software ADI file to reinstall on the CSS.

To format a disk, use the following commands:

- **format** *disk_slot* - Formats the specified disk. The slot number designates which disk you want to format. Valid *disk_slot* selections are 0 (for the disk in slot 0) or 1 (for the disk in slot 1).
- **format** *disk_slot* {**quick**} - Formats the specified disk (0 or 1). The quick option reformats the disk without performing cluster verification.

Note Use the quick disk format only when you are certain of the disk integrity.

For example, to format the disk in slot 1, enter:

```
# format 1
```

The CSS queries you about formatting the disk.

```
Formatting the disk results in all disk data being
permanently erased.
Are you sure you want to continue? (yes,no):
```

Enter one of the following:

- **yes** to reformat the disk.
- **no** to end the reformat function. If the disk has unrecoverable errors and you do not reformat it, be aware that the file system may be corrupt and functionality is compromised.

Specifying a Disk for Booting, Logging, and Core Dumps

Use the **map** commands to specify the disk (slot 0 or slot 1) that the CSS uses to store the primary boot record, the secondary boot record, logging output file, and core dumps. By default, disk 0 is the default storage location for the primary and secondary boot records in the CSS. The default storage location for log files and core dumps is the specified disk from which the CSS boots (disk 0 or disk 1).

You can mix and match the storage location of these files between the two disks. For example, you can store the primary boot record on disk 0 and the secondary boot record on disk 1, and redirect the storage of output logs and core dumps to disk 1.

The syntax for this global configuration mode command is:

map [core|log|primary-boot|secondary-boot] *disk_slot*

The options for the **map** command are as follows:

- **core** - Specifies the disk that contains the core dumps
- **log** - Specifies the disk that contains the logging output
- **primary-boot** - Specifies the disk that contains the primary boot record
- **secondary-boot** - Specifies the disk that contains the secondary boot record

Use the **no** form of each command to remove mapping to the specified disk and return the setting to the default disk.

Selecting a Disk for the Primary Boot Record

Use the **map primary-boot** command to select the disk that contains the primary boot record of the CSS. Disk 0 is the default storage location for the primary boot record 0. Valid selections are 0 (for the disk in slot 0) and 1 (for the disk in slot 1). This command is available only in SuperUser mode.

For example, to select the disk in slot 1 as the storage location for the primary boot record, enter:

```
# map primary-boot 1
```

To return the storage location of the primary boot record back to the disk in slot 0, enter:

```
# no map primary-boot
```

or

```
# map primary-boot 0
```

Selecting a Disk for the Secondary Boot Record

Use the **map secondary-boot** command to select the disk that contains the secondary boot record of the CSS. Disk 0 is the default storage location for the secondary boot record. Valid selections are 0 (for the disk in slot 0) and 1 (for the disk in slot 1). This command is available only in SuperUser mode.

For example, to select the disk in slot 1 as the storage location for the secondary boot record, enter:

```
# map secondary-boot 1
```

To return the storage location of the secondary boot record back to the disk in slot 0, enter:

```
# no map secondary-boot
```

or

```
# map secondary-boot 0
```

Selecting a Disk for Core Dumps

Use the **map core** command to select the disk that stores core dump files when the CSS experiences a fatal error. The default storage location for core dump files is the disk from which the CSS boots (disk 0 or disk 1). For example, if the CSS boots from disk 1, then disk 1 becomes the default storage location for core dump files.

Valid selections are 0 (disk in slot 0) and 1 (disk in slot 1). This command is available only in SuperUser mode.

Note Core dump information is intended for Customer Support use only.

For example, to select the disk in slot 1 as the storage location for core dumps, enter:

```
# map core 1
```

To return the storage location for core dumps back to boot disk, enter:

```
# no map core
```

Selecting a Disk for Logging

Use the **map log** command to select the disk on which you want to store log files. The default storage location for log files is the disk from which the CSS boots (disk 0 or disk 1). For example, if the CSS boots from disk 0, then disk 0 becomes the default storage location for log files.

Valid selections are 0 (disk in slot 0) and 1 (disk in slot 1). This command is available only in SuperUser mode.

Note Logging to a CSS disk can cause the performance of the CSS to degrade. If logging requires frequent writes to disk (that is, several hundred log messages per day), we recommend that you log to a hard disk and store all other system files on a Flash disk. Although Flash disks generally provide the most reliable way to store information over time, hard disks endure frequent writes to disk better than the Flash disks currently available.

For example, to select the disk in slot 1 as the storage location for log files, enter:

```
# map log 1
```

To return the storage location of log files back to the boot disk, enter:

```
# no map log
```

Copying Files Between Disks

Use the **copy** command to copy the startup configuration, logs, scripts, archive, and boot image files from one disk (source) to the second disk (destination) in a CSS. The CSS software automatically creates the software directory and hierarchy on the destination disk. This command is available only in SuperUser mode.

The syntax is:

copy *source_disk_slot* {**log** *filename* {*destination filename*}|**logs**|**script** *filename* {*destination filename*}|**scripts**|**archive** *filename* {*destination filename*}|**archives**|**boot-image** *filename*|**startup-config**}

The options and variables for the **copy** command are as follows:

- *source_disk_slot* - Specifies the disk location containing the files you want to copy. Valid entries are 0 (disk in slot 0) and 1 (disk in slot 1). If you want to perform a complete copy of all contents from the source disk to the second disk, enter only the *disk_slot* value. Do not enter values for the additional **copy** command variables.
- **log** *filename* - Copies the specified log file from the source disk to the second disk.
- **log** *filename* {*destination filename*} - Copies the specified log file from the source disk to the second disk using a different destination filename.
- **logs** - Copies all log files from the source disk to the second disk.
- **script** *filename* - Copies the specified script from the source disk to the second disk.
- **script** *filename* {*destination filename*} - Copies the specified script from the source disk to the second disk using a different destination filename.
- **scripts** - Copies all scripts from the source disk to the second disk.
- **archive** *filename* - Copies the specified archive file from the source disk to the second disk.
- **archive** *filename* {*destination filename*} - Copies the specified archive file from the source disk to the second disk using a different destination filename.
- **archives** - Copies all archive files from the source disk to the second disk.
- **boot-image** *filename* - Copies the specified boot image ADI from the source disk to the second disk. If necessary, use the **show installed-software** command to view the names of the boot-images (see the "Showing Software Information" section for details on using the **show installed-software** command).
- **startup-config** - Copies the startup-config file from the source disk to the second disk.

Note the following restrictions for the **copy** command when copying information between two disks in the CSS:

- The source file must exist.
- An equivalent release of CSS software must be present on the destination disk before you copy information to the disk (such as a startup-config file, a log file, or a script). If necessary, copy the boot image to the second disk before copying a startup-config file, log file, or script.

Showing the Disk Mapping Configurations

Use the **show map** command to display the mapping configuration of the two disks in slot 0 and slot 1 in a CSS. This command displays the disk assignment of the primary-boot record, the secondary-boot record, core dump files, and logging output. This command is available in all modes.

For example:

```
(config)# show map

MSD Mapping:
Primary-Boot:   0
Secondary-Boot: 0
Core:           1
Log:            1
```

Showing Files from a Disk

Use the **show** command to display the specified archive, log, script, or startup configuration file stored on a specific disk in the CSS. The syntax is:

show *disk_slot* {**log** *filename*|**script** *filename*|**archive** *filename*|
startup-config}

The options and variables for the **show** command are as follows:

- *disk_slot* - Specifies the disk location containing the file to display. The valid entries are 0 (disk in slot 0) and 1 (disk in slot 1).
- **log** *filename* - Displays the contents of a log (or trap log file) from the specified disk.
- **script** *filename* - Displays the contents of the script from the specified disk.
- **archive** *filename* - Displays the contents of the archive filename from the specified disk.
- **startup-config** - Displays the contents of the CSS startup configuration file from the specified disk.

Clearing Files from a Disk

Use the **clear** command to delete the specified file (startup configuration, logs, scripts, archive file) stored on a specific disk in the CSS. This command is available only in SuperUser mode. The syntax is:

clear *disk_slot* {**log** *filename*|**script** *filename*|**archive** *filename*|
startup-config}

The options and variable for the **clear** command are as follows:

- *disk_slot* - Specifies the disk location containing the file to delete. Valid entries are 0 (disk in slot 0) and 1 (disk in slot 1).
- **log** *filename* - Deletes the specified log (or trap log file) from the disk.
- **script** *filename* - Deletes the specified script from the disk.
- **archive** *filename* - Deletes the specified archive filename from the disk.
- **startup-config** - Deletes the CSS startup configuration file from the disk.

Unpacking and Removing an ADI

Note Before unpacking the ADI, you must first copy the ADI to the CSS disk. Use the **copy ftp ftp_record** *filename* **boot-image** command to copy the ADI to the CSS disk. Refer to Chapter 4, Specifying the CSS Boot Configuration for details.

Use the **unpack** command to unpack the ArrowPoint Distribution Image (ADI) on the CSS disk. Enter the ADI filename as an unquoted text string with a maximum of 32 characters. For example:

```
(config-boot)# unpack ap0720002.adi
```

Use the **remove** command to remove an ArrowPoint Distribution Image (ADI) that is not currently running on the CSS. For a dual-disk CSS, you need to identify the specified disk.

Warning **Ensure you do not delete the software version that you are currently running in the CSS.**

To remove a software version installed on the CSS, use the following commands:

- **remove** *software version* **-** Enter the ADI filename as an unquoted text string with a maximum of 32 characters.
- **remove** *disk_slot software version* - Enter the slot location of the disk (0 or 1) in a dual disk CSS, followed by the ADI filename as an unquoted text string with a maximum of 32 characters.

To display a list of ADIs installed on your CSS, enter **remove ?**. To display the ADI you are currently running, use the **version** command.

To remove an ADI, enter:

```
(config-boot)# remove ap0720001
```

To remove an ADI from a disk in slot 1 of a dual-disk CSS, enter:

```
(config-boot)# remove ap0720001 1
```

Archiving Files to the Archive Directory

Use the **archive** command and options to archive files. Archiving is useful when you update software and want to save a script, log, or startup-config file from a previous release of software. The archive directory on the CSS disk stores the archive files.

The syntax for this command is:

> **archive** [[**startup-config**|**log** *log_filename*|**script** *script_filename*]
> *archive_filename*}|**running-config** *archive_filename*]

The options for this command are as follows:

- **archive startup-config** - Archives the startup-config file
- **archive log** - Archives a log file
- **archive script** - Archives a script file
- **archive running-config** - Archives the running-config file

To display the contents of the archive directory, enter **show archive ?**. Archive files include running-config and startup-config files, scripts, and user profiles.

You must archive your startup-config file and scripts before you upgrade the CSS software or these files will be overwritten during the upgrade. Once the CSS completes the upgrade and reboots, use the **restore** command to copy these files from the archive directory as the current startup-config file and scripts.

This section includes the following topics:

- Archiving the Startup-Config File
- Archiving a Log File
- Archiving Scripts
- Archiving the Running-Config File

Note If you booted your CSS from a network-mounted system and your hard drive does not work, the CSS suspends all archive-related functions.

Archiving the Startup-Config File

Use the **archive startup-config** command to archive the startup-config file. Enter the archive filename as an optional name you want to assign to the archive file. Enter an unquoted text string with a maximum of 32 characters. The syntax for this command is:

archive startup-config {*archive_filename*}

Archiving a Log File

Use the **archive log** command to archive a log file. The syntax for this command is:

archive log *log_filename* **{***archive_filename***}**

The variables are as follows:

- *log_filename* - The filename of the log to archive. To see a list of log files, enter **archive log ?**.
- *archive_filename* - (Optional) The name you want to assign to the archive file. Enter an unquoted text string with a maximum of 32 characters.

Archiving Scripts

Use the **archive script** command to archive a script file. The syntax for this command is:

archive script *script_filename* {*archive_filename*}

The variables are as follows:

- *script_filename* - The filename of the script to archive. To see a list of scripts, enter **archive script ?**.
- *archive_filename* - (Optional) The name you want to assign to the archive file. Enter an unquoted text string with a maximum of 32 characters.

Archiving the Running-Config File

Use the **archive running-config** command to archive the running-config file. Enter the archive filename as the name you want to assign to the archive file. The archive filename is an unquoted text string with a maximum of 32 characters. The syntax for this command is:

archive running-config *archive_filename*

Note You can also use the **save_config** alias command to automatically copy the running-config to the startup-config, and then archive the startup-config.

Clearing the Archive Directory

Use the **clear archive** command to clear a file in the archive directory. Enter the archive filename as the name of the archive file to clear. To list the archive files, enter **clear archive ?**. The syntax for this command is:

clear archive *archive_filename*

Restoring Files from the Archive Directory

Use the **restore** command to restore files previously archived in the CSS archive directory. The archive directory on the CSS disk stores log, script, and startup configuration files. The archive directory resides on the CSS disk (hard or Flash disk).

The syntax for this command is:

restore *archive_filename* [**log** {*log_filename*} |**script**
{*script_filename*}|**startup-config**]

The options for this command are as follows:

- **restore** *archive_filename* **log** - Restores an archived log file to the log subdirectory.
- **restore** *archive_filename* **script** - Restores an archived script file to the script subdirectory.
- **restore** *archive_filename* **startup-config** - Restores an archived startup-config file to the startup configuration.

This section includes the following topics:

- Restoring an Archived Log File
- Restoring an Archived Script File
- Restoring an Archived Startup-Config File

Note If you booted your CSS from a network-mounted system and your hard drive does not work, the CSS suspends all restore-related functions.

Restoring an Archived Log File

Use the **restore log** command to restore an archived log file to the log subdirectory. The syntax for this command is:

restore *archive_filename* **log** {*log_filename*}

The variables are as follows:

- *archive_filename* - The name of the archived log file. Enter an unquoted text string. To see a list of archived files, enter **restore ?**.
- *log_filename* - (Optional) The name you want to assign to the restored log file. Enter an unquoted text string with a maximum of 32 characters.

For example, to restore the log file *arrowlog* to the log subdirectory and rename the log file to *arrowpointlog*, enter:

```
# restore arrowlog log arrowpointlog
```

Restoring an Archived Script File

Use the **restore** *archive_filename* **script** command to restore an archived script file to the script subdirectory. The syntax for this command is:

restore *archive_filename* **script** {*script_filename*}

The variables are as follows:

- *archive_filename* - The name of the archived file. Enter an unquoted text string. To see a list of archived files, enter **restore ?**.
- *script_filename* - (Optional) The name you want to assign to the script file. Enter an unquoted text string with a maximum of 32 characters.

For example, to restore the script *arrowscript* to the script subdirectory, enter.

```
# restore arrowscript script
```

Restoring an Archived Startup-Config File

Use the **restore** *archive_filename* **startup-config** command to restore an archived file to the startup configuration.

Caution The restored file overwrites the startup configuration.

The syntax for this command is:

restore *archive_filename* **startup-config**

Enter the archived startup-config filename as an unquoted text string. To see a list of archived files, enter **restore ?**.

For example, to restore the archived startup-config file *arrowstart* as the current startup-config file, enter:

```
# restore arrowstart startup-config
```

Enabling and Copying Core Dumps

A core dump occurs when the CSS experiences a fatal error. The CSS allows you to enable or disable core dumps. Core dumps are enabled by default.

When the CSS experiences a fatal error and core dumps are enabled, the CSS:

- Writes information about the fatal error to the Core directory of the volume root (for example, c:\core) on either the hard or Flash disk. The CSS stores one dump file per slot for each card type until the disk (Flash or hard disk) is full. Files can be 10 to 20 MB in size.
- Reboots automatically.

Note Core dump information is for Cisco Technical Assistance Center (TAC) use only.

When the CSS experiences a fatal error and core dumps are disabled, the CSS reboots automatically. The CSS does not write information to the hard disk or the Flash disk.

For a Flash disk-based system, if the core dump file is older than 15 minutes, the file may be overwritten. If you want to save the core dump file for later examination, archive the file to another directory or disk before it is overwritten. For details on using the **archive log** command, refer to the "Archiving the Startup-Config File" section.

This section includes the following topics:

- Enabling and Disabling Core Dumps
- Showing Core Dumps
- Copying Core Dumps to an FTP or TFTP Server

Enabling and Disabling Core Dumps

To disable core dumps, enter:

```
(config)# dump disable
```

To reenable core dumps (the default setting), enter:

```
(config)# dump enable
```

Showing Core Dumps

Use the **show core** command to display the core dump files stored in the Core directory of the volume root (for example, c:\core) on the hard disk or Flash disk. This command is available in all modes except User mode.

Use the **show core** *disk_slot* command to display the core dump files stored in the Core directory of the volume root of a specific disk in the CSS. Valid selections are 0 (for the disk in slot 0) or 1 (for the disk in slot 1).

For example:

```
# show core

SCP0101_4.80_115... OCT 31 15:06:26       16708412
SCP0101_4.80_109... OCT 29 16:56:16       37806459
SCP0101_4.80_116... NOV  1 15:54:28       38403870
```

Copying Core Dumps to an FTP or TFTP Server

Use the **copy core** command to copy core dumps from the CSS to a File Transfer Protocol (FTP) or Trivial File Transfer Protocol (TFTP) server. This command is available in SuperUser mode. A core dump occurs when the CSS experiences a fatal error.

To see a list of core dumps, enter the **copy core ?** command.

Note The CSS FTP server supports only the active (normal) FTP mode of operation. It does not support the passive FTP mode of operation.

Copying Core Dumps to an FTP Server

Use the **copy core ftp** command to copy a core dump to an FTP server. This command is available only in SuperUser mode.

Before you copy a core dump from the CSS to an FTP server, create an FTP record file containing the FTP server IP address, username, and password. For information on configuring an FTP record, see the "Creating an FTP Record" section.

The syntax for this command is:

copy core *coredump_filename* **ftp** *ftp_record filename*

The variables are as follows:

- *coredump_filename* - The name of the core dump on the CSS. Enter an unquoted text string with no spaces and a maximum of 32 characters.
- *ftp_record* - The name of the FTP record file that contains the FTP server IP address, username, and password. Enter an unquoted text string with no spaces and a maximum of 32 characters.
- *filename* - The name you want to assign to the file on the FTP server. Include the full path to the file. Enter an unquoted text string with no spaces and a maximum of 32 characters.

For example:

```
# copy core dumpfile ftp ftpserv1 starlogthurs
```

Copying Core Dumps to a TFTP Server

Use the **copy core tftp** command to copy a core dump to an TFTP server. This command is available only in SuperUser mode.

The syntax for this command is:

copy core *coredump_filename* **tftp** *ip_address_hostname filename*

The variables are as follows:

- *coredump_filename* - The name of the core dump on the CSS. Enter an unquoted text string with no spaces and a maximum of 32 characters.
- *ip_address_hostname* - The IP address or host name of the TFTP server to receive the file. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1) or in mnemonic host-name format (for example, myhost.mydomain.com). If you wish to use a host name, you must first set up a host table using the **host** command.
- *filename* - The name you want to assign to the file on the TFTP server. Include the full path to the file. Enter an unquoted text string with no spaces and a maximum of 32 characters.

Showing CSS Configurations

The CSS CLI provides a comprehensive set of **show** commands that display CSS configurations. The **show** commands are mode-independent; that is, they are available in each mode. The CSS does not show configuration default values in the individual show output, even when you specify a CLI command to configure a default value.

To display the list of **show** commands, enter: `(config)# show ?`

This section includes the following topics:

- Showing Software Information
- Showing Hardware Information
- Showing System Resources
- Showing System Uptime
- Showing Disk Information
- Showing User Information
- Showing Current Logins

Showing Software Information

To display the software versions installed on the CSS, use the following commands:

- **show version** - Displays details about the current installed software version, including the version of Flash software code, whether the software is set to primary or secondary, and your license number.
- **show installed-software version-limit** - Displays the maximum number of software versions allowed on your CSS.
- **show installed-software -** Displays a list of currently installed software on the CSS.
- **show installed-software** *disk_slot* - Displays a list of currently installed software on a specific disk in a dual-disk CSS. Valid selections are 0 (for the disk in slot 0) or 1 (for the disk in slot 1).

Note Use the **version** command in SuperUser mode to display the version of software currently running on the CSS. This display also shows the version of Flash software code, whether the software is set to primary or secondary, and your license number.

For example:

```
# show version
Version:            ap0720001 (7.20 Build 1)
Network Path:       e:/adi_directory/
Config Path:        e:/adi_directory/
Flash (Locked):     7.20 Build 1
Flash (Operational):7.20 Build 2
Type:               PRIMARY
License Cmd Set(s): Standard Feature Set
                    Enhanced Feature Set
                    SSH Server
```

Showing Hardware Information

Use the **show chassis** command to display a chassis configuration for the CSS. The syntax and options for this command are as follows:

- **show chassis** - Displays a summary of the chassis configuration.
- **show chassis slot** *number* - Displays the operational parameters for a slot in a CSS 11503 or CSS 11506 chassis. Enter an integer value for the chassis slot number.
- **show chassis verbose** - Displays detailed information about the chassis configuration.
- **show chassis flash** - Displays the operational and locked Flash software code on the CSS 11501, and the CSS 11503 or CSS 11506 SCM and I/O modules. An asterisk (*) character before a Flash version of code and build number indicates that it is active.
- **show chassis inventory** - Displays the physical configuration of the CSS including part and serial numbers.
- **show chassis session-processors** - Displays the weight and power summary of the session processors in the CSS chassis.

For example, to view a summary of the CSS chassis configuration, enter:

```
# show chassis
```

Table 3-1 describes the fields in the **show chassis** command output.

Table 3-1 Field Descriptions for the show chassis Command

Field	Description
Product Name	The model number of the CSS.
SW Version	The software version currently running on the CSS.
Serial Number	The serial number of the chassis Flash memory device.
Base MAC Address	The MAC address for the chassis.
Slot/Module Number	The number of the CSS 11501, CSS 11503, or CSS 11506 chassis slot in which the module resides.
Module Name	The name of the module installed in the CSS.
Status	The operational status of the module. The possible states are as follows: • primary • backup • powered-off • powered-on • bad • unknown • empty slot
Slot/Port	The slot and port number on the CSS 11503 or CSS 11506 (for example, 2/1).
Port Number	The port number on the CSS 11501 (for example, 1).
Name	The name of the interface port on the CSS 11501 or the module installed in the CSS 11503 or CSS 11506.
Status	The operational status of the interface port/module. The possible states are as follows: • Online • Offline

Table 3-2 describes the fields in the **show chassis slot** command output.

Table 3-2 Field Descriptions for the show chassis slot Command

Field	Description
Product Name	The model number of the CSS.
SW Version	The software version currently running on the CSS.
Serial Number	The serial number of the chassis Flash memory device.
Base MAC Address	The MAC address for the chassis.
Slot Number	The number of the CSS 11503 or CSS 11506 chassis slot in which the module resides.
Type	The name and product number of the installed module.
Serial Number	The serial number of the module.
Number of Ports	The total number of ports in an I/O module.
Status	The operational status of the module. The possible states are as follows: • primary • backup • powered-off • powered-on • bad • unknown • empty slot
Port Number	The Ethernet port number.

Table 3-2 Field Descriptions for the show chassis slot Command (continued)

Field	Description
Port Name	The port name.
Operational Status	The status of the port. The possible states are as follows: • online • offline-ok • offline-bad • bad • going-online • going-offline • inserted • post • post-ok • post-fail • post-bad-comm • any • unknown-state

Table 3-3 describes the fields in the **show chassis verbose** command output.

Table 3-3 Field Descriptions for the show chassis verbose Command

Field	Description
Product Name	The model number of the CSS.
SW Version	The software version currently running on the CSS.
Serial Number	The serial number of the chassis Flash memory device.
Base MAC Address	The MAC address for the chassis.
Module(s) Found	The number of modules installed in the chassis.
Power Supplies Found	The number of power supplies installed in the chassis.
Fan(s) Found	The number of fans installed in the chassis.
Slot/Subslot	The number of the CSS 11503 or CSS 11506 chassis slot in which the module resides.
Module Name	The name of the module installed in the CSS 11501.
Operational	The active Flash code on the CSS.
Locked	The inactive Flash code available on the CSS.
Slot Number	The number of the CSS 11503 or CSS 11506 chassis slot in which the module resides.
Module Number	The number of the CSS 11501 chassis slot in which the module resides.
Type	The name and product number of the installed module.
Serial Number	The serial number of the module.
Number of Ports	The total number of ports in an I/O module.

Table 3-3 Field Descriptions for the show chassis verbose Command (continued)

Field	Description
Status	The operational status of the module. The possible states are as follows: • primary • backup • powered-off • powered-on • bad • unknown • empty slot
Port Number	The Ethernet port number.
Port Name	The port name.
Operational Status	The status of the port. The possible states are as follows: • online • offline-ok • offline-bad • bad • going-online • going-offline • inserted • post • post-ok • post-fail • post-bad-comm • any • unknown-state

Table 3-4 describes the fields in the **show chassis flash** command output.

Table 3-4 Field Descriptions for the show chassis flash Command

Field	Description
Product Name	The model number of the CSS.
SW Version	The currently running software version on the CSS.
Serial Number	The serial number of the chassis Flash.
Base MAC Address	The MAC address for the chassis.
Slot/Subslot	The number of the CSS 11503 or CSS 11506 chassis slot in which the module resides.
Module Name	The name of the module installed in the CSS 11501.
Operational	The active Flash code on the CSS.
Locked	The inactive Flash code available on the CSS.

Table 3-5 describes the fields in the **show chassis inventory** command output.

Table 3-5 Field Descriptions for the show chassis inventory Command

Field	Description
Product Name	The model number of the CSS.
SW Version	The software version currently running on the CSS.
Serial Number	The serial number of the chassis Flash memory device.
Base MAC Address	The MAC address for the chassis.
Slot	The number of the CSS 11503 or CSS 11506 chassis slot in which the module resides.
Module	The number of the CSS 11501 chassis slot in which the module resides.
Part	The name of the board in the CSS 11501 chassis.
Module/Part Name	The name of the module installed in the CSS.
Serial	The serial number of the module.

Table 3-6 describes the fields in the **show chassis session-processors** command output.

Table 3-6 Field Descriptions for the show chassis session-processor Command

Field	Description
Chassis Total Weight	The combined relative weights of all active session processors in the CSS chassis.
SP Modules Total/Active	The total number of installed modules that contain session processors, and the number of active modules that contain session processors.
Name	The name of the module installed in the CSS.
Slot	The number of the CSS 11503 or CSS 11506 chassis slot in which the module resides.
Module	The number of the CSS 11501 chassis slot in which the module resides.
Slot	For a CSS 11503 or CSS 11506, the number of the chassis slot in which the session processor resides.
Sub	For a CSS 11503 or CSS 11506, the number of the chassis module subslot in which the session processor resides.
Weight	A value assigned to an SP based on its ability to provide session processing. An active SP has a relative weight assignment greater than 0. A weight of 0 prevents the SP from performing any session processing.
Power Percentage (%)	A value calculated from an SP-assigned weight relative value that represents the session processor share of the total session processing capacity in the chassis.

Showing System Resources

Use the **show system-resources** command to display information about the size of the installed memory and free memory available on the:

- CSS 11501.
- CSS 11503 or CSS 11506 SM and SCM module. The CSS displays system resources for the primary SCM.

Table 3-7 describes the fields in the **show system-resources** command output.

Table 3-7 Field Descriptions for the show system-resources Command

Field	Description
Installed Memory	The total memory size in the CSS
Free Memory	The amount of free memory available
CPU	The utilized percentage of the CPU
Buffer Statistics	
Buffer Pool	The buffer pool index
Size	The size, in bytes, of each data buffer in the buffer pool
Total	The total number of buffers in the buffer pool
Available	The current number of available buffers in the buffer pool
Failures	The number of failures to obtain a buffer from the buffer pool
Low Buffer Count	The lowest recorded number of available buffers

Use the **show system-resources cpu_summary** command to display a summary of the CPU utilization by all modules in the CSS 11501, CSS 11503, or CSS 11506 chassis.

Table 3-8 describes the fields in the **show system-resources cpu_summary** command output.

Table 3-8 Field Descriptions for the show system-resources cpu_summary Command

Field	Description
Name/Module	The name of the module installed in the CSS.
Slot	For a CSS 11503 or CSS 11506, the number of the chassis slot in which the module resides.
Sub	For a CSS 11503 or CSS 11506, the number of the chassis module subslot in which the memory resides.
Module	The number of the module in the CSS 11501 chassis.
CPU%	The percentage of the total CPU capacity that is currently in use.

Showing System Uptime

Use the **show uptime** command to display the length of time the CSS has been running. The time is displayed in *hour*:*minute*:*second* format. For the CSS 11503 or CSS 11506, this command shows the length of time each module has been running.

To display how long the CSS has been running, enter:

```
# show uptime
Uptime:
10 days 03:25:22
```

Showing Disk Information

Use the **show disk** command to view general information about the CSS hard disk or Flash disk. The information includes the total number of clusters on the disk, the free space available, and the number of files, folders, and bad clusters on the disk.

To display specific CSS disk information, use the following **show disk** commands:

- **show disk** - Displays disk information for the hard disk or Flash disk. If the CSS includes two disks, the **show disk** command lists information for both disks.
- **show disk** *disk_slot* - Displays disk information for a specific slot in a dual-disk CSS. Valid selections are 0 (for the disk in slot 0) or 1 (for the disk in slot 1). The default is the disk from which the CSS booted.

For example, to display CSS disk information for the disk in slot 1, enter:

```
# show disk 1
```

Table 3-9 describes the fields in the **show disk** command output for the CSS.

Table 3-9 Field Descriptions for the show disk Command

Field	Description
Total # of Clusters	The total number of clusters on the disk
Bytes Per Cluster	The number of bytes in each cluster
Free Clusters	The number of available clusters on the disk
Bad Clusters	The number of bad clusters on the disk
Free Bytes	The available disk space, in bytes and megabytes
Max Contiguous Free Bytes	The maximum number of contiguous free bytes (and megabytes) found on the disk
Files	The number of files on the disk
Folders	The number of folders on the disk
Total Bytes in Files	The total number of bytes in all of the files found on the disk

Table 3-9 Field Descriptions for the show disk Command (continued)

Field	Description
Lost Chains	The total number of lost chains found on the disk
Total Bytes in Lost Chains	The total number of bytes in all of the lost chains found on the disk

Showing User Information

Use the **show user-database** command to view CSS operating information related to a single user, or to multiple users. This command displays user information related to login privileges, the type of user, and directory access privileges.

To display all users currently defined in the CSS, enter:

```
(config)# show user-database
```

To display information for a specific user, enter:

```
(config)# show user-database picard
```

Table 3-10 describes the fields in the **show user-database** command output.

Table 3-10 Field Descriptions for the show user-database Command

Field	Description
Virtual Authentication	Identifies if users must enter a username and password to log in to the CSS.
Console Authentication	Identifies if console port authentication of locally defined usernames and passwords logging in to the CSS in enabled.
Username	The name of the user.
Privilege Level	The privilege level of the user.

Table 3-10 Field Descriptions for the show user-database Command (continued)

Field	Description
Type	The type of user. Types are as follows: • Administrator (administrative username, created using the **username-offdm** command) • Technician (technician username, created using the **username-technician** command) If the field is blank, the user is neither an administrator nor a technician. **Note** The **username-offdm** command is for use by system administrative personnel only. The **username-technician** command is for use by technical personnel only.
Directory Access	The directory access privileges for the listed usernames (as specified through the **dir-access** option of the **username** command). There are a series of access privilege codes assigned to the seven CSS directories in the following order: Script, Log, Root (installed CSS software), Archive, Release Root (configuration files), Core, and MIBs directories. By default, users have both read- and write-access privileges (B) to all seven directories. The levels for each of the CSS directories can be one of the following access privilege codes: • R - Read-only access to the CSS directory • W - Write-only access to the CSS directory • B - Both read- and write-access privileges to the CSS directory (default for all users) • N - No access privileges to the CSS directory For example, BBNBNBB indicates that the user has no access to the root and release root directories, but has read and write access to the script, log, archive, core, and MIB directories.

Showing Current Logins

Use the **show lines** command to display currently connected lines or sessions. A connected line is a console or Telnet session. This command is available in all modes.

To display currently connected lines or sessions, enter:

```
(config)# show lines
```

Table 3-11 describes the fields in the **show lines** output.

Table 3-11 Field Descriptions for the show lines Command

Field	Description
Line	The type of session. The * indicates your current session.
User	The login name of the user.
Login	The amount of time that the user has been logged in on the CSS.
Idle	The amount of time that the session has been idle.
Location	The location where the session is occurring.

Where to Go Next

Chapter 4, Specifying the CSS Boot Configuration provides information on how to setup the boot configuration for the CSS, including configuring an FTP record and specifying the primary and secondary location from which the CSS accesses the boot image.

Specifying the CSS Boot Configuration

This chapter describes how to set the boot configuration, both the primary and secondary boot files, for the CSS. Information in this chapter applies to all CSS models, except where noted.

This chapter contains the following major sections:

- Boot Setup Quick Start
- Accessing Boot Mode
- Specifying the Primary Boot Configuration
- Specifying the Secondary Boot Configuration
- Configuring a Boot Configuration Record for the Passive SCM
- Showing the Boot Configuration
- Booting the CSS from a Network Drive

As an alternate procedure for managing the CSS boot configuration from the CLI, you can use the Offline DM menu. Refer to Appendix B, Using the Offline Diagnostic Monitor Menu for details.

Boot Setup Quick Start

Table 4-1 provides a quick overview of the steps required to configure the CSS to boot from a primary boot file and from a secondary boot file. 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, see the sections following Table 4-1.

Table 4-1 Boot Setup Quick Start

Task and Command Example
1. Create a File Transfer Protocol (FTP) record file to use when accessing an FTP server from the CSS. This step is optional. `# ftp-record arrowrecord 192.168.19.21 bobo password "secret" /outgoing` **Note** Refer to Chapter 3, Managing the CSS Software for details on creating an FTP record.
2. Access boot mode. `(config)# boot`

3. Specify the primary boot configuration.
`(config-boot)# primary boot-file ap0720002`
`(config-boot)# primary boot-type boot-via-ftp arrowrecord`

4. Specify the secondary boot configuration.
`(config-boot)# secondary boot-file ap0720001`
`(config-boot)# secondary boot-type boot-via-disk`

5. Exit from boot mode. `(config-boot)# exit`
6. Save your configuration changes to the startup-config file (recommended). If you do not save the running configuration, all configuration changes are lost upon reboot. `(config)# copy running-config startup-config`

Accessing Boot Mode

Boot configuration mode contains all the commands necessary to boot the CSS and maintain the software revision. To access this mode, use the **boot** command from global configuration mode.

To access boot mode, enter:

```
(config)# boot
```

The CSS enters boot mode.

```
(config-boot)#
```

Specifying the Primary Boot Configuration

Use the **primary** command to specify the primary boot configuration. The options for this boot-mode command are as follows:

- **primary boot-file** - Specifies the primary boot file
- **primary boot-type** - Specifies the primary boot method: local disk, using FTP, or a network-mounted file system using FTP
- **primary config-path** - Specifies the path to a network CSS configuration

This section includes the following topics:

- Specifying the Primary Boot File
- Specifying the Primary Boot Type
- Specifying the Primary Configuration Path

Specifying the Primary Boot File

Use the **primary boot-file** command to specify the primary boot file. Enter the primary boot file as an unquoted text string with no spaces and a maximum of 64 characters.

To specify the primary boot filename, enter:

```
(config-boot)# primary boot-file ap0720002
```

To display a list of boot filenames, enter:

```
(config-boot)# primary boot-file ?
```

To remove the primary boot file, enter:

```
(config-boot)# no primary boot-file
```

Specifying the Primary Boot Type

Use the **primary boot-type** command to specify the location from which the CSS accesses the primary boot image upon system reboot or when you download new software.

The syntax for this boot mode command is:

primary boot-type [**boot-via-disk**|**boot-via-ftp** *ftp_record*|
    boot-via-network *ftp_record*]

The options and variables for this command are as follows:

- **boot-via-disk** - Boots the CSS from a software version that resides on the CSS disk.
- **boot-via-ftp** *ftp_record* - Downloads an ADI file containing CSS software that you want to install on the CSS disk. The CSS accesses the ADI or GZIP file containing the CSS software from an FTP server, copies the file to the disk, and unpacks it. The CSS then boots from the disk.
- **boot-via-network** *ftp_record* - Uses FTP to boot the CSS from software located on a network-mounted file system on a remote system. Instead of the CSS disk, the network file system contains the CSS software. The CSS boots from this file system and loads the configuration in to memory.

Note A network boot requires that the CSS contains an operational disk.

The *ftp_record* variable is the name of the FTP record file that contains the FTP server IP address, username, and password. Enter an unquoted text string with no spaces. Refer to Chapter 3, Managing the CSS Software for details on creating an FTP record.

For example, to configure the primary boot-type to **boot-via-disk**, enter:

```
(config-boot)# primary boot-type boot-via-disk
```

To remove the primary boot type, enter:

```
(config-boot)# no primary boot-type
```

Primary Boot Configuration Considerations

When you select **primary boot-type boot-via-ftp** or **primary boot-type boot-via-network**, make sure you properly connect the Ethernet Management port on the CSS to the network. The locations of the Ethernet Management port on the CSS are listed below.

- CSS 11501 - Front Panel 10 Mbps-Ethernet Management port
- CSS 11503 and CSS 11506 - SCM 10 Mbps-Ethernet Management port

When you select **primary boot-type boot-via-network**, make sure you:

- Locate the remote system on the network where you will copy the CSS software.
  - Make sure the CSS can access the system via FTP.
  - Copy the CSS software Zip file from www.cisco.com onto the designated network server.
  - Create a directory and unzip the file in to the directory. This directory will contain all of the boot files and directories.
- Create an FTP record on the CSS to the directory that contains the CSS software on the network drive. Refer to Chapter 3, Managing the CSS Software for details on creating an FTP record.

Note Be aware of the following network boot restrictions: a network boot is not supported on UNIX workstations, and the War-FTP daemon is not supported for network-booting the system software.

A network boot requires that the CSS contains an operational disk.

Specifying the Primary Configuration Path

Use the **primary config-path** command to specify the alternate path to a network configuration for the network boot method. An alternate configuration path allows multiple CSSs to use the same boot image while keeping their configuration information in separate directories. The CSS must be able to access the configuration path through an FTP server as defined in the FTP record for the network boot method.

When using an alternate configuration path, make sure the path leads to a directory containing the script, log, and information subdirectories, and to the startup-config file. These subdirectories must contain the files in the corresponding subdirectories of the unzipped boot image. First, create these subdirectories on the FTP server, then copy the files from the boot image to the subdirectories.

Enter the configuration pathname as an unquoted text string with no spaces and a maximum of 64 characters.

To configure the primary configuration path, enter:

```
(config-boot)# primary config-path f:/bootdir/
```

To remove the primary network configuration path, enter:

```
(config-boot)# no primary config-path
```

Specifying the Secondary Boot Configuration

Use the **secondary** command to specify the secondary boot configuration. The CSS uses the secondary boot configuration when the primary boot configuration fails. The options for this boot mode command are as follows:

- **secondary boot-file** - Specifies the secondary boot file
- **secondary boot-type** - Specifies the boot method: local disk or FTP
- **secondary config-path** - Specifies the path to a network configuration using FTP

This section includes the following topics:

- Specifying the Secondary Boot File
- Specifying the Secondary Boot Type
- Specifying the Secondary Configuration Path

Specifying the Secondary Boot File

Use the **secondary boot-file** command to specify the secondary boot file that the CSS uses when the primary boot configuration fails. Enter the boot file as an unquoted text string with no spaces and a maximum of 64 characters.

To specify the secondary boot filename, enter:

```
(config-boot)# secondary boot-file ap0720001
```

To display a list of secondary boot filenames, enter:

```
(config-boot)# secondary boot-file ?
```

To remove the secondary boot file, enter:

```
(config-boot)# no secondary boot-file
```

Specifying the Secondary Boot Type

Use the **secondary boot-type** command to specify the secondary boot configuration.

The syntax for this boot mode command is:

secondary boot-type [**boot-via-disk**|**boot-via-ftp** *ftp_record*|
boot-via-network *ftp_record*]

The options and variables for this command are as follows:

- **boot-via-disk** - Boots the CSS from a software version that resides on the CSS disk.
- **boot-via-ftp** *ftp_record* - Downloads an ADI file containing CSS software that you want to install on the CSS disk. The CSS accesses the ADI or GZIP file containing the CSS software from an FTP server, copies the file to the disk, and unpacks it. The CSS then boots from the disk.
- **boot-via-network** *ftp_record* - Uses FTP to boot the CSS from software located on a network-mounted file system on a remote system. Instead of the CSS disk, the network file system contains the CSS software. The CSS boots from this file system and loads the configuration in to memory.

Note A network boot requires that the CSS contains an operational disk.

The *ftp_record* variable is the name of the FTP record file that contains the FTP server IP address, username, and password. Enter an unquoted text string with no spaces. Refer to Chapter 3, Managing the CSS Software for details on creating an FTP record.

For example, to specify the secondary boot type as **boot-via-disk**, enter:

```
(config-boot)# secondary boot-type boot-via-disk
```

To remove the secondary boot type, enter:

```
(config-boot)# no secondary boot-type
```

Secondary Boot Configuration Considerations

When you select **secondary boot-type boot-via-ftp** or **secondary boot-type boot-via-network**, make sure you properly connect the Ethernet Management port on the CSS to the network. Note the locations of the Ethernet Management port on the CSS as listed below.

- CSS 11501 - Front Panel 10 Mbps-Ethernet Management port
- CSS 11503 and CSS 11506 - SCM 10 Mbps-Ethernet Management port

When you select **secondary boot-type boot-via-network**, make sure you:

- Locate the remote system on the network where you will copy the CSS software.
  - Make sure the CSS can access the system via FTP.
  - Copy the CSS software Zip file from www.cisco.com onto the designated network server.
  - Create a directory and unzip the file in to the directory. This directory will contain all of the boot files and directories.
- Create an FTP record on the CSS to the directory that contains the CSS software on the network drive.

Note Be aware of the following network boot restrictions: a network boot is not supported on UNIX workstations, and the War-FTP daemon is not supported for network-booting the system software.

A network boot requires that the CSS contains an operational disk.

Specifying the Secondary Configuration Path

Use the **secondary config-path** command to specify the alternate path to a network configuration for the network boot method. An alternate configuration path allows multiple CSSs to use the same boot image while keeping their configuration information in separate directories. The CSS must be able to access the configuration path through an FTP server as defined through the FTP record for the network boot method.

When using an alternate configuration path, make sure the path leads to a directory containing the script, log, and information subdirectories, and to the startup-config file. These subdirectories must contain the files in the corresponding subdirectories of the unzipped boot image. First, create these subdirectories on the FTP server, then copy the files from the boot image to the subdirectories.

Enter the configuration pathname as an unquoted text string with no spaces and a maximum of 64 characters.

To configure the secondary configuration path, enter:

```
(config-boot)# secondary config-path f:/bootdir/
```

To remove the secondary network configuration path, enter:

```
(config-boot)# no secondary config-path
```

Configuring a Boot Configuration Record for the Passive SCM

Use the **passive** command to configure the boot configuration record for the current passive SCM installed in a CSS 11506. The boot configuration record consists of the IP address, subnet mask, boot method, and boot file.

Using the **sync** options for this command, copy the boot configuration record from the active SCM to the passive SCM. In most CSS configurations, the active and passive SCMs have the same boot record.

The **passive** command also allows you to configure the individual components of the boot configuration record on the passive SCM. For example, you can configure a boot record on the passive SCM that has a software version that differs from the active SCM. The boot configuration record allows you to run a new software version on the active SCM and have an older software version on the passive SCM.

You can also configure a different IP address on the passive SCM to track an active-to-passive state transition between the SCMs. You can track active-to-passive state transitions through a network management station, where you can receive SNMP host traps.

The **passive** command and its options affect only the current passive SCM. When you configure the passive SCM, the set values are loaded in to its NVRAM. If the passive SCM transitions to the active state, it continues to retain these values, but is no longer affected by these commands; boot commands are not saved in the running-config file.

This section includes the following topics:

- Configuring the Passive SCM Gateway Address
- Configuring the Passive SCM IP Address
- Configuring the Passive SCM Primary Boot File
- Configuring the Passive SCM Primary Boot Type
- Configuring the Passive SCM Primary Configuration Path
- Configuring the Passive SCM Secondary Boot File
- Configuring the Passive SCM Secondary Boot Type
- Configuring the Passive SCM Secondary Configuration Path

- Configuring the Passive SCM Subnet Mask
- Copying Configuration Information from the Active SCM to the Passive SCM

Configuring the Passive SCM Gateway Address

Use the **passive gateway address** command to configure an Ethernet management port default gateway to load a boot file on a CSS across different subnets for the passive SCM. Enter the IP address for the passive SCM to be used upon CSS boot up. Do not enter an all-zero IP address.

For example:

```
(config-boot)# passive gateway address 172.16.3.6
```

To change the passive SCM boot gateway address, reenter the **passive gateway address** command.

Configuring the Passive SCM IP Address

Use the **passive ip address** command to configure the boot IP address for the passive SCM. Enter the IP address for the passive SCM to be used upon CSS boot up. Do not enter an all-zero IP address.

For example:

```
(config-boot)# passive ip address 172.16.3.6
```

To change the passive SCM boot IP address, reenter the **passive ip address** command.

Configuring the Passive SCM Primary Boot File

Use the **passive primary boot-file** command to specify the primary boot image for the passive SCM. Enter the filename of the primary boot image for the passive SCM as an unquoted text string with no spaces and a maximum of 64 characters. To display a list of filenames, enter **passive primary boot-file ?**.

For example:

```
(config-boot)# passive primary boot-file ap0720002
```

To remove the primary boot file from the passive SCM, enter:

```
(config-boot)# no passive primary boot-file
```

Configuring the Passive SCM Primary Boot Type

Use the **passive primary boot-type** command to specify the location from which the CSS accesses the primary boot image for the passive SCM upon system reboot or when you download new software.

The syntax for this boot mode command is:

passive primary boot-type [boot-via-disk|boot-via-ftp *ftp_record*|
boot-via-network *ftp_record*]

The options and variables for the **passive primary boot-type** are as follows:

- **boot-type boot-via-disk** - Boots the CSS from a software version that currently resides on the CSS disk.
- **boot-type boot-via-ftp** *ftp_record* - Downloads an ADI file containing CSS software that you want to install on the CSS disk. The CSS accesses the ADI or GZIP file containing the CSS software from an FTP server, copies the file to the disk, and unpacks it. The CSS then boots from the disk.
- **boot-type boot-via-network** *ftp_record* - Uses FTP to boot the CSS from software located on a network-mounted file system on a remote system. Instead of the CSS disk, the network file system contains the CSS software. The CSS boots from this file system and loads the configuration in to memory.

Note A network boot requires that the CSS contains an operational disk.

The *ftp_record* variable is the name of the FTP record file that contains the FTP server IP address, username, and password. Enter an unquoted text string with no spaces. Refer to Chapter 3, Managing the CSS Software for details on creating an FTP record.

For example:

```
(config-boot)# passive primary boot-type boot-via-ftp arecord
```

To remove the primary boot type from the passive SCM, enter:

```
(config-boot)# no passive primary boot-type
```

Configuring the Passive SCM Primary Configuration Path

Use the **passive primary config-path** command to specify the alternate path to a network configuration for the network boot method for the passive SCM. An alternate configuration path allows multiple CSSs to use the same boot image while keeping their configuration information in separate directories. The CSS must be able to access the configuration path through an FTP server as defined through the FTP record for the network boot method.

When using an alternate configuration path, make sure the path leads to a directory containing the script, log, and information subdirectories, and the startup-config file. These subdirectories must contain the files in the corresponding subdirectories in the unzipped boot image. First, create these subdirectories on the FTP server, then copy the files from the boot image to the subdirectories.

Enter the configuration path for network configuration. Enter an unquoted text string with no spaces and a maximum of 64 characters. For example:

```
(config-boot)# passive primary config-path c:/bootdir/
```

To remove the primary network configuration path, enter:

```
(config-boot)# no passive primary config-path
```

Configuring the Passive SCM Secondary Boot File

Use the **passive secondary boot-file** command to specify the secondary boot image for the passive SCM. Enter the name of the boot file for the primary boot image as an unquoted text string with no spaces and a maximum of 64 characters. To display a list of boot filenames, enter **passive secondary boot-file ?**. For example:

```
(config-boot)# passive secondary boot-file ap0720001
```

To remove the secondary boot file from the passive SCM, enter:

```
(config-boot)# no passive secondary boot-file
```

Configuring the Passive SCM Secondary Boot Type

Use the **passive secondary boot-type** command to specify the secondary boot configuration for the passive SCM. The secondary boot configuration is used when the primary configuration fails.

The syntax for this boot mode command is:

passive secondary boot-type [**boot-via-disk**|**boot-via-ftp** *ftp_record*|
boot-via-network *ftp_record*]

The options and variables for the **passive secondary boot-type** command are as follows:

- **boot-type boot-via-disk** - Boots the CSS from a software version that resides on the CSS disk.
- **boot-type boot-via-ftp** *ftp_record* - Downloads an ADI file containing CSS software that you want to install on the CSS disk. The CSS accesses the ADI or GZIP file containing the CSS software from an FTP server, copies the file to the disk, and unpacks it. The CSS then boots from the disk.
- **boot-type boot-via-network** *ftp_record* - Uses FTP to boot the CSS from software located on a network-mounted file system on a remote system. Instead of the CSS disk, the network file system contains the CSS software. The CSS boots from this file system and loads the configuration in to memory.

Note A network boot requires that the CSS contains an operational disk.

The *ftp_record* variable is the name of the FTP record file that contains the FTP server IP address, username, and password. Enter an unquoted text string with no spaces. Refer to Chapter 3, Managing the CSS Software for details on creating an FTP record.

For example:

```
(config-boot)# passive secondary boot-type boot-via-disk
```

To remove the secondary boot type from the passive SCM, enter:

```
(config-boot)# no passive secondary boot-type
```

Configuring the Passive SCM Secondary Configuration Path

Use the **passive secondary config-path** command to specify the secondary alternate path to a network configuration for the network boot method for the passive SCM. An alternate configuration path allows multiple CSSs to use the same boot image while keeping their configuration information in separate directories. The CSS must be able to access the configuration path through an FTP server as defined through the FTP record for the network boot method.

When using an alternate configuration path, make sure that the path leads to a directory containing the script, log, and information subdirectories, and the startup-config file. These subdirectories must contain the files in the corresponding subdirectories of the unzipped boot image. First, create these subdirectories on the FTP server, then copy the files from the boot image to the subdirectories.

Enter the configuration path as an unquoted text string with no spaces and a maximum of 64 characters.

For example:

```
(config-boot)# passive secondary config-path c:/bootdir/
```

To remove the primary network configuration path, enter:

```
(config-boot)# no passive secondary config-path
```

Configuring the Passive SCM Subnet Mask

Use the **passive subnet mask** command to configure the system boot subnet mask for the passive SCM.

For example:

```
(config-boot)# passive subnet mask 255.255.0.0
```

Copying Configuration Information from the Active SCM to the Passive SCM

Use the **passive sync** command to copy the primary and secondary boot configuration record from NVRAM of the active SCM to the passive SCM. For the CSS 11506, the **passive sync** command also copies the startup- configuration file and synchronizes the clock time from the active SCM to the passive SCM. This command is available in boot mode.

To synchronize specific boot configuration, startup configuration, or clock time information between the active SCM and the passive SCM in a CSS 11506, use the following commands:

- **passive sync boot-config** - Copies the boot configuration record from the active SCM to the passive SCM.
- **passive sync startup-config** - Copies the startup-config file from the active SCM to the passive SCM.
- **passive sync image** - Copies the ADI of the boot-image file from the active SCM to the passive SCM.
- **passive sync time** - Synchronizes the clock time of the passive SCM with the active SCM.

To copy the primary and secondary boot configuration record, startup configuration, and clock time on a CSS 11506, enter:

```
(config-boot)# passive sync
```

To copy the boot configuration record from the active SCM to the passive SCM in a CSS 11506, enter:

```
(config-boot)# passive sync boot-config
```

Showing the Boot Configuration

Use the **show boot-config** command to display the boot configuration. For example:

```
(config-boot)# show boot-config

!*********************** BOOT CONFIG ***********************
ip address 172.16.36.58
subnet mask 255.0.0.0
primary boot-file ap0720001
primary boot-type boot-via-disk
```

Booting the CSS from a Network Drive

Network booting enables you to boot the CSS from a network drive using a .zip file of the CSS software version located on www.cisco.com. When you configure the CSS for network boot, the CSS must contain an operational disk (hard or Flash).

Use your customer login and password to access www.cisco.com. From this location, you can access the page listing the network boot .zip file versions of CSS software. Click an image to download the software.

Note Be aware of the following network boot restrictions: a network boot is not supported on UNIX workstations, and the War-FTP daemon is not supported for network-booting the system software. In addition, network booting does not support the use of core dumps from the CSS.

Perform a network boot if you want multiple CSSs to use the same boot image while keeping their own configuration information. Provide an alternate path for the location of the configuration information. This information must exist on the same network file system as the boot image.

When using an alternate configuration path, make sure the path leads to a directory containing the script, log, and information subdirectories. These subdirectories must contain the files in the corresponding subdirectories in the boot image. Create these subdirectories, then copy the files from the boot image.

This section includes the following topics:

- Configuring Network Boot for a Primary SCM
- Configuring Network Boot for a Passive SCM
- Showing Network Boot Configurations

Configuring Network Boot for a Primary SCM

To configure network boot for a primary SCM on the CSS 11503 or CSS 11506:

1. Make sure the SCM management port has access to the network drive from which you are booting the CSS. The SCM mounts the drive, and reads and writes to the network drive.
2. Use FTP to install the software .zip file to the network drive base directory specified in the FTP record. This network directory must be the same directory that you use to boot the CSS.
3. Unzip the file. You must use the .zip distribution format for network loading.
4. Configure the FTP record. Refer to Chapter 3, Managing the CSS Software for details on creating an FTP record. Note that the config-path and the base directory path in the FTP record associated with the network boot must contain a pathname that is distinct from a non-network drive name (for example, c: or host:).

For example:

```
# ftp-record bootrecord 192.168.19.21 bobo encrypted-password
"secret" e:/adi_directory/
```

This directory must contain the unzipped files.

5. Configure the CSS to boot from a network drive. For example:

```
(config-boot)# primary boot-type boot-via-network bootrecord
```

6. Optionally, configure a primary configuration path to allow multiple CSSs to use the same boot image while keeping their configuration information in separate directories. The CSS must be able to access the configuration path through the FTP server as defined in the FTP record. For example:

```
(config-boot)# primary config-path e:/adi_directory/
```

Configuring Network Boot for a Passive SCM

To configure network boot for a passive SCM on the CSS 11503 or CSS 11506:

1. Configure an FTP record for the passive SCM, if not already configured (see the "Configuring a Boot Configuration Record for the Passive SCM" section).

2. Make sure the passive SCM management port has access to the network drive from which you are booting the CSS. If the primary SCM fails, the passive SCM connects to the remote disk and loads the software configuration.

3. Configure the CSS to boot from a network drive. For example:

```
(config-boot)# passive primary boot-type boot-via-network
bootrecord
```

To display a list of configured FTP records, reenter the command and specify the **?** character. For example:

```
(config-boot)# passive primary boot-type boot-via-network
bootrecord ?
```

4. Optionally, configure a primary configuration path to allow multiple CSSs to use the same boot image while keeping their configuration information in separate directories. Your FTP daemon must support the drive mapping. Also, the CSS must be able to access the configuration path through the FTP server as defined in the FTP record. For example:

```
(config-boot)# primary config-path e:/adi_directory/
```

Showing Network Boot Configurations

Use the **show version** command to display the network boot configuration. For example:

```
(config)# show version

Version:            ap0720001 (7.20 Build 1)
Network Path:       e:/adi_directory/
Config Path:        e:/adi_directory/
Flash (Locked):     7.20 Build 1
Flash (Operational):7.20 Build 2
Type:               PRIMARY
Licensed Cmd Set(s):Standard Feature Set
                    Enhanced Feature Set
                    Secure Management
```

Note Use the **version** command in SuperUser mode to display the network boot configuration.

To display network boot configuration information, use the **show boot-config** command. For example:

```
(config)# show boot-config

!*********************** BOOT CONFIG ***********************
secondary config-path  e:/adi_directory/
secondary boot-type    boot-via-network Secondary-Boot
primary boot-file      ap0720001
primary boot-type      boot-via-network
subnet mask            255.0.0.0
ip address             192.168.4.226
```

Where to Go Next

Chapter 5, Configuring Interfaces and Circuits provides information on how to configure the CSS interfaces and circuits and how to bridge interfaces to VLANs.

CHAPTER 5

Configuring Interfaces and Circuits

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

This chapter contains the following major sections:

- Interface and Circuit Overview
- Configuring Interfaces
- Configuring Circuits
- Configuring RIP for an IP Interface
- Configuring the Switched Port Analyzer Feature

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 assign the Ethernet interfaces to a specific 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 Ethernet interface port (Fast Ethernet port or Gigabit Ethernet port). A trunk is a point-to-point link carrying the traffic of several VLANs. The advantage of a trunk is to save ports by creating a link between two CSSs implementing VLANs. 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.

With trunking enabled, the CSS automatically inserts a tag in every frame transmitted over the trunk link to identify the originating VLAN. When the VLAN-aware CSS receives the frame, it 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.

For an 802.1Q trunk, you can use the **default-vlan** command to:

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

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

Figure 5-1 illustrates the interfaces, circuits, and VLANs in a CSS, and Figure 5-2 illustrates trunking between VLANs.

Figure 5-1 CSS Interfaces and Circuits

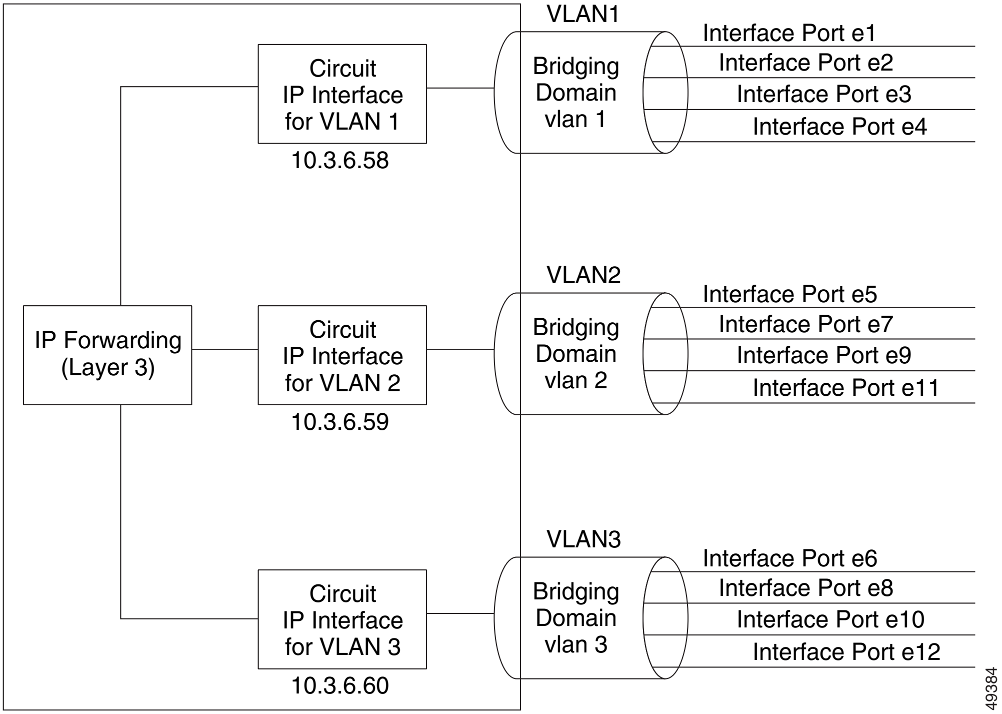
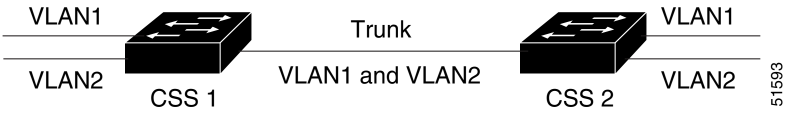


Figure 5-2 Interface Trunking Between VLANs



Interface and Circuit Configuration Quick Start

Table 5-1 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, see the sections following Table 5-1.

Table 5-1 Interface and Circuit Configuration Quick Start

Task and Command Example
1. Log in to the CSS.

2. Enter configuration mode by typing **config**.
`# config`
`(config)#`

3. Enter the interface mode for the interface you wish to configure.
This set of interface commands applies to the CSS 11501.
`(config)# interface e1`
`(config-if[e1])#`
This set of interface commands applies to the CSS 11503 and CSS 11506.
`(config)# interface 2/1`
`(config-if[2/1])#`

4. Configure the interface duplex, speed, and flow control (default is **auto-negotiate**).
`(config-if[2/1])# phy auto-negotiate disable`
`(config-if[2/1])# phy 1Gbits-FD-sym`

5. Bridge the interface to a VLAN. All interfaces are assigned to VLAN1 by default. `(config-if[2/1])# bridge vlan 2`

6. Enable trunking for a CSS Gigabit Ethernet or Fast Ethernet port (optional).
`(config-if[2/1])# trunk`
`(config-if[2/1])# vlan 2`
`Create VLAN<2>, [y/n]:y`
`(config-if-vlan[2/1-2])# vlan 3`
`Create VLAN<3>, [y/n]:y`
`(config-if-vlan[2/1-3])#`

Table 5-1 Interface and Circuit Configuration Quick Start (continued)

Task and Command Example

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

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

8. Display the interface configuration (optional).

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

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

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

10. Display the circuit configuration (optional).

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

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

    ```
    (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 11501 differ slightly from the commands to configure interfaces on the CSS 11503 and CSS 11506 because they require a slot/port designation. The CSS 11501 does not use the slot/port designation.

This section includes the following topics:

- Configuring an Interface
- Entering a Description for the Interface
- Configuring Interface Duplex and Speed
- Setting Interface Maximum Idle Time
- Bridging an Interface to a VLAN
- Specifying VLAN Trunking for an Interface
- Configuring Spanning-Tree Bridging for a VLAN or a Trunked Interface
- Configuring Port Fast on an Interface
- Showing Interface Configurations
- Shutting Down an Interface
- Shutting Down All Interfaces
- Restarting an Interface
- Restarting All Interfaces

Configuring an Interface

Use the **interface** command to configure an Ethernet interface. Enter the interface name as follows:

- CSS 11501 - Enter the interface name in *interface port* format (for example, e1 for Ethernet interface port 1).
- CSS 11503 or CSS 11506 - Enter the interface format in *slot/port* format (for example, 3/1 for Ethernet port 1 on the I/O module in slot 3).

For example, to configure interface port 1 on a CSS 11501, access interface mode for the port by entering:

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

For example, to configure interface 1 on a CSS 11503 or CSS 11506, access interface mode for the I/O module in slot 2 by entering:

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

Note in both examples that the CSS changes from configuration mode to the specific interface mode.

Entering a Description for the Interface

Use the **description** command to identify the Ethernet interface. Enter a quoted text string from 1 to 255 characters including spaces.

For example:

```
(config-if[2/1])# description "Connects to server17"
```

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

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

!************************ INTERFACE ************************
interface 2/1
   description "Connects to server17"
   bridge vlan 2
```

To remove an interface description, enter:

```
(config-if[2/1])# no description
```

Configuring Interface Duplex and Speed

By default, the CSS Fast Ethernet interface and Gigabit Ethernet interface are configured to auto-negotiate. The CSS automatically detects the network line speed (Fast Ethernet only) and duplex of incoming signals, and synchronizes those parameters during data transfer. Auto-negotiation enables the CSS and the other devices on the link to achieve the maximum common level of operation.

When using Fast Ethernet ports with older equipment that cannot transmit the duplex and speed with the signals, you can manually configure the speed (10 Mbps, 100 Mbps) and duplex (half or full duplex) of the CSS port to match the transmitting equipment.

When you use Gigabit Ethernet ports, if the link does not come up (perhaps due to traffic congestion), you may need to force the CSS and its link partner in to a specific mode. The CSS allows you to manually select a full duplex and flow control (pause frame) mode. Flow control allows the CSS to control traffic during congestion by notifying the other port to stop transmitting until the congestion clears. When the other device receives the pause frame, it temporarily stops transmitting data packets. When the CSS detects local congestion and becomes overwhelmed with data, the Gigabit Ethernet ports transmits a pause frame. Both the CSS Gigabit Ethernet and its link partner must be configured with the same pause method (asymmetric, symmetric, or both). By default, all Gigabit Ethernet ports are configured to full duplex mode with symmetric pause (pause frames transmitted and received by the CSS).

Use the **phy** command to configure the duplex, speed (Fast Ethernet ports only), and flow control (Gigabit Ethernet ports only) for the interface ports, as follows:

- **phy auto-negotiate** - Resets the Fast Ethernet and Gigabit Ethernet ports to automatically negotiate port speed and duplex of incoming signals.

Note Pause mode during auto-negotiation is not supported for the Fast Ethernet ports.

- **phy auto-negotiate** {**enable** | **disable**} - Disables the Gigabit Ethernet interface from automatically negotiating duplex of incoming signals. By default, auto-negotiation is enabled for all Gigabit Ethernet ports.

  Gigabit Ethernet port auto-negotiation remains enabled when a pause mode command is specified so the Gigabit Ethernet interface ports can act upon the link partner's flow control capability. If it is necessary to disable auto-negotiation for the Gigabit Ethernet port when using a pause mode, enter the **phy auto-negotiate disable** command.
- **phy 10Mbits-FD** - Sets the Fast Ethernet port to 10 Mbps and full-duplex mode.
- **phy 10Mbits-HD** - Sets the Fast Ethernet port to 10 Mbps and half-duplex mode.
- **phy 100Mbits-FD** - Sets the Fast Ethernet port to 100 Mbps and full-duplex mode.
- **phy 100Mbits-HD** - Sets the Fast Ethernet port to 100 Mbps and half-duplex mode.
- **phy 1Gbits-FD-asym** - Sets the Gigabit Ethernet port to full-duplex mode with asymmetric pause frames transmitted toward the link partner. Asymmetric pause is useful when you need the CSS to pause its link partner but not to respond to pause frames transmitted from the link partner.
- **phy 1Gbits-FD-no pause** - Sets the Gigabit Ethernet port to full-duplex mode with no pause frames transmitted or received.
- **phy 1Gbits-FD-sym** - Sets the Gigabit Ethernet port to full-duplex mode with symmetric pause (pause frames transmitted and received by the CSS). Symmetric pause is useful for point-to-point links. By default, all Gigabit Ethernet ports are configured to full-duplex mode with symmetric pause.
- **phy 1Gbits-FD-sym-asym** - Sets the Gigabit Ethernet port to full-duplex mode with symmetric and asymmetric pause frames used with the local device.

For example, to configure Fast Ethernet interface 1 on the I/O module in slot 2 of the CSS 11503 to 100 Mbps and half-duplex mode, enter:

```
(config-if[2/1])# phy 100Mbits-HD
```

For example, to configure gigabit interface 1 on the SCM in slot 1 of the CSS 11503 to full-duplex mode with asymmetric pause, enter:

```
(config-if[1/1])# phy auto-negotiate disable
(config-if[1/1])# phy 1Gbits-FD-asym
```

Setting 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 the interface automatically.

Set the idle time to a value greater than the interval over which the interface is 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 an idle time from 15 to 65535 seconds. The default is 0, which disables the idle timer.

For example, to set the maximum idle time to 180 seconds for interface port 1 on a CSS 11503, the I/O module in slot 2, enter:

```
(config-if[2/1])# max-idle 180
```

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

```
(config-if[2/1])# no max-idle
```

Bridging an Interface to a VLAN

Use the **bridge vlan** command to specify a VLAN and associate it with the specified Ethernet interface. Enter an integer from 1 to 4094 as the VLAN identifier. The default is 1. All interfaces are assigned to VLAN1 by default.

The following list defines the maximum number of VLANs supported by the specific CSS models:

- CSS 11501 and CSS 11503 - A maximum of 256 VLANs
- CSS 11506 - A maximum of 512 VLANs

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

For example, to configure e1 to VLAN2 on the CSS 11501, enter:

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

The CSS Gigabit Ethernet and Fast Ethernet interface ports support trunking to multiple VLANs through the **trunk** command. In this configuration, use the **trunk** command for the Ethernet interface instead of the **bridge vlan** command (and the other associated bridge CLI commands). See "Showing Interface Configurations" for details.

To restore the default VLAN1 on the CSS 11501, enter:

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

To display all interfaces and the VLANs to which they are configured, use the **show circuit** command. In the **show circuit** display, VLANs appear as VLAN (uppercase, with no space before the VLAN number). See the "Showing Circuits" section for information about the **show circuits** command.

Specifying VLAN Trunking for an Interface

Use the **trunk** command to activate VLAN trunking for a CSS interface. You specify all VLANs that include the specified port as part of the VLAN. The **trunk** command also converts the link in to a trunk link. Use the **vlan** command to specify the number of each VLAN to be associated with the Gigabit Ethernet or Fast Ethernet port. Enter an integer from 1 to 4094 as the VLAN identifier.

The following list defines the maximum number of VLANs supported by the specific CSS models:

- CSS 11501 and CSS 11503 - A maximum of 256 VLANs
- CSS 11506 - A maximum of 512 VLANs

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 Ethernet or Fast Ethernet 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.

When you specify the **trunk** command, enter the word **vlan** in lowercase letters and include a space before the VLAN number (for example, **vlan 2**). The CSS automatically prompts you to create the specified VLAN (where **y** instructs the software to create the VLAN and **n** cancels the VLAN creation).

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

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

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 disable trunking on the specified interface and associated VLANs, enter:

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

To display all interfaces and the VLANs to which they are configured, use the **show circuit** command. In the **show circuit** output, VLANs appear as VLAN (uppercase, with no space before the VLAN number). For an interface that has trunking enabled, an "-*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. See the "Showing Circuits" section for information about the **show circuits** command.

Selecting a Default VLAN in a Trunk

To define a default VLAN 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 are discarded.

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

For example:

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

To remove the default VLAN selection, enter:

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

Configuring Spanning-Tree Bridging for a VLAN or a Trunked Interface

The CSS supports configuration of Spanning-Tree Protocol (STP) bridging for an Ethernet interface in a VLAN or for a trunked Ethernet interface. Spanning-tree bridging is used to detect, and then prevent, loops in the network. You can define the bridge spanning-tree path cost, priority, and state for an Ethernet interface or for a trunked Ethernet interface. Ensure you configure the spanning-tree bridging parameters the same on all switches running STP in the network.

Note When connecting a Cisco Catalyst switch to a CSS using an 802.1Q trunk and the Spanning-Tree Protocol, the Catalyst runs a spanning-tree instance for each VLAN. When you configure an 802.1Q trunk on an Ethernet interface for the Catalyst switch, the bridge protocol data units (BPDUs) are tagged with the corresponding VLAN ID and the destination MAC address changes from the standard 01-80-C2-00-00-00 to the proprietary 01-00-0c-cc-cc-cd. This modification allows Cisco switches operating in a non-Cisco (a mix of other vendors) 802.1Q trunk environment to maintain spanning-tree states for all VLANs. Although the CSS maintains a spanning-tree instance for each VLAN as well, the CSS uses the standard 01-80-C2-00-00-00 destination MAC address for all BPDUs (tagged or untagged). When you connect a Cisco Catalyst switch to a CSS over an 802.1Q trunk, the result is that neither switch recognizes the other's BPDUs, and both assume root status. If a spanning-tree loop is detected, the Catalyst switch goes in to blocking mode on one of its looped ports.

This section includes the following topics:

- Configuring Spanning-Tree Bridge Pathcost
- Configuring Spanning-Tree Bridge Priority
- Configuring Spanning-Tree Bridge State

For details about globally configuring spanning-tree bridging paremeters for the CSS (such as bridge aging time, forward delay time, hello time interval, and maximum age), refer to Chapter 6, Configuring CSS Network Protocols.

Configuring Spanning-Tree Bridge Pathcost

Use the **bridge pathcost** command to set the spanning-tree path cost for an Ethernet interface or for a trunked Ethernet interface. The cost is the contribution of the interface to the vast path cost towards the 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 on the CSS 11501, enter:

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

For example, to set a path cost of 2 for the I/O module in slot 1, Ethernet port 1, in VLAN3, enter:

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

To restore the default path cost, enter:

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

Configuring Spanning-Tree Bridge Priority

Use the **bridge priority** command to set the spanning-tree bridge priority for an Ethernet interface or for a trunked Ethernet interface. If the CSS has a bridge priority that is lower than all other switches, it will be automatically selected by the other switches as the root switch. Enter an integer from 0 to 255. The default is 128.

For example, to set a bridge priority of 100 for e7 on the CSS 11501, enter:

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

For example, to set a bridge priority of 100 for the I/O module in slot 1, Ethernet port 1, in VLAN3, enter:

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

To restore the default priority of 128, enter:

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

Configuring Spanning-Tree Bridge State

Use the **bridge state** command to set the spanning-tree bridge state for an Ethernet interface or for a trunked Ethernet interface. By default, an Ethernet interface is set to the enabled bridge state.

For example, to enable the bridge state for e7 on the CSS 11501, enter:

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

For example, to enable the bridge state for the I/O module in slot 1, Ethernet port 1, in VLAN3, enter:

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

To disable the bridge state, enter:

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

Configuring Port Fast on an Interface

The Port Fast feature immediately brings a CSS Ethernet interface (port) to the Spanning Tree Protocol (STP) forwarding state from a blocking state, bypassing the listening and learning states. You can specify Port Fast for ports connected to a single workstation or server to allow those devices to immediately connect to the network, rather than waiting for the STP to converge.

Ports connected to a single workstation or server should not receive bridge protocol data units (BPDUs).

Caution The purpose of Port Fast is to minimize the time ports must wait for STP to converge. This means that the Port Fast function is effective only when used on ports connected to end stations in the network. If you enable Port Fast on a port connecting to another switch, you risk creating a spanning-tree loop. Consider using the BDPU guard feature to avoid creating a spanning-tree loop.

This section includes the following topics:

- Enabling Port Fast
- Enabling BPDU Guard
- Showing Port Fast Information

Enabling Port Fast

A port with the Port Fast feature enabled is moved directly to the spanning-tree forwarding state without waiting for the standard forward-time delay.

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

To enable Port Fast on a non-trunked port, use the interface mode **bridge port-fast enable** command. You cannot configure Port Fast on a trunked port. By default, Port Fast is disabled on the port.

```
(config-if[2/1])# bridge port-fast enable
```

To disable the Port Fast feature, use the interface mode **bridge port-fast disable** command.

```
(config-if[2/1])# bridge port-fast disable
```

Enabling BPDU Guard

Use the BPDU guard feature to prevent a Port Fast port on the CSS from participating in the spanning tree. When you globally enable BPDU guard on the Port Fast ports, spanning tree shuts down the ports that receive BPDUs. For information to enable Port Fast on an interface port, see the "Configuring Port Fast on an Interface" section.

In a valid configuration, the enabled Port Fast ports do not receive BPDUs. Receiving a BPDU on a Port Fast-enabled port signals an invalid configuration, such as the connection of an unauthorized device, and the BPDU guard feature puts the port in the disabled state. The BPDU guard feature provides a secure response to invalid configurations because you must manually put the port back in service.

To enable the BPDU guard on the CSS, use the global configuration **bridge bdpu-guard enabled** command:

```
#(config) bridge bpdu-guard enabled
```

To disable BPDU guard, use the global configuration **bridge bpdu-guard disabled** command:

```
#(config) bridge bpdu-guard disabled
```

Showing Port Fast Information

To display whether Port Fast is enabled or disabled on all interfaces, use the **show bridge port-fast** command. This command is available in all modes. This command also displays whether the BPDU guard feature is enabled or disabled on the CSS, and the state of the interfaces.

Table 5-2 describes the fields in the **show bridge port-fast** command output.

Table 5-2 Field Description for the show bridge port-fast Command

Field	Description
BPDU guard is *state* on this switch.	The state of the BPDU guard feature on the CSS: Enabled or Disabled.
Name	The number of the module slot and interface.
IfIndex	The interface index number.
Type	The type of interface. • **fe** indicates a Fast Ethernet interface. • **ge** indicates a Gigabit Ethernet interface.
Oper	The operational state of the interface: Up or Down.
Admin	The administration state: Enable or Down.
PortFast State	Indicates whether Port Fast is enabled or disabled on the interface.

Showing Interface Configurations

This CSS includes a series of **show** interface mode commands that enable you to view interface configuration information about the CSS. This information includes VLAN bridging, VLAN trunk status, list of valid Ethernet interfaces, interface duplex and speed values, interface statistics, and errors on an Ethernet interface.

This section includes the following topics:

- Showing Bridge Configurations
- Showing Trunking Configurations
- Showing Interface Information
- Showing Interface Duplex and Speed
- Showing Interface Statistics
- Showing Ethernet Interface Errors

Showing Bridge Configurations

The CSS enables you to show bridging information for a specific VLAN in the CSS. Use the **show bridge** command to display this bridging information.

The syntax for this command is:

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

The options and variables are as follows:

- **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 Spanning Tree Protocol (STP) state; designated root; bridge ID; 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**] **?**

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

Table 5-3 describes the fields in the **show bridge forwarding** command output.

Table 5-3 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 bridge forwarding table

Table 5-4 describes the fields in the **show bridge status** command output.

Table 5-4 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, during which the host times out root information.
Root Hello Time	The interval, in seconds, that the root bridge broadcasts its hello message to other CSSs.
Root Fwd Delay	The delay time, in seconds, that the root bridge 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.

Table 5-4 Field Descriptions for the show bridge status Command (continued)

Field	Description
State	The state of the port. The possible states are as follows: • Block - The blocking state. A port enters the blocking state after CSS 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 nonoperational.
Designated Bridge	The bridge ID for the designated bridge.
Designated Root	The bridge ID for the designated root.
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 Gigabit Ethernet and Fast Ethernet ports. To display this information, use the **show trunk** command.

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

Table 5-5 Field Descriptions for the show trunk Command

Field	Description
Port	The CSS port
VLAN	The VLAN on the port
Default VLAN	The configured default VLAN on the port (if there is no configured default VLAN, “None” appears in this field)

Showing Interface Information

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

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

To display information for a specific interface, enter the **show interface** command and the interface name. Enter the interface name as follows:

- CSS 11501 - Enter the interface name in *interface port* format (for example, e1 for Ethernet interface port 1).
- CSS 11503 or CSS 11506 - Enter the interface format in *slot/port* format (for example, 3/1 for Ethernet port 1 on the I/O module in slot 3).

For example, to show interface information for port 1 on a CSS 11503, the I/O module in slot 2, enter:

```
(config)# show interface 2/1
```

Table 5-6 describes the fields in the **show interface** command output.

Table 5-6 Field Descriptions for the show interface Command

Field	Description
Name	The name of the interface.
ifIndex	The Index for the interface.
Type	The type of interface. The possible types include: • fe - Fast Ethernet interface • ge - Gigabit Ethernet interface • console - Console interface
Oper	Operational state: Up or Down.
Admin	Administrative state: Up or Down.
Last Change	The date of the last state change.

Showing Interface Duplex and Speed

Use the **show phy** command to show duplex and speed values for all interfaces. For example:

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

To show duplex and speed value for a specific interface, specify the **show phy** command and the interface name. Enter the interface name as follows:

- CSS 11501 - Enter the interface name in *interface port* format (for example, e1 for Ethernet interface port 1).
- CSS 11503 or CSS 11506 - Enter the interface format in *slot/port* format (for example, 3/1 for Ethernet port 1 on the I/O module in slot 3).

For example, to show the interface and duplex speed for interface port 1 on a CSS 11506, the I/O module in slot 2, enter:

```
(config)# show phy 2/1
```

Table 5-7 describes the fields in the **show phy** command output.

Table 5-7 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 the speed is automatically negotiated.
Configured Duplex	The configured duplex for the Ethernet interface (port) in the CSS. Auto indicates the duplex is automatically negotiated.
Actual Speed	The actual speed for the Ethernet interface (port) in the CSS.
Actual Duplex	The configure duplex for the Ethernet interface (port) in the CSS.
Link	The link status: Up or Down.
Rev	Revision number of the chip.
Partner Auto	Indicates whether auto-negotiation is available on the link partner.

Showing Interface Statistics

Use the **show mibii** command to display the extended 64-bit MIB-II statistics for a specific interface, or for all interfaces in the CSS. The CSS Enterprise ap64Stats MIB defines these statistics. The Gigabit Ethernet module port statistics are an aggregation of all ports on the module.

To display the RFC 1213 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 ?**.

Note Refer to Chapter 12, Configuring Simple Network Management Protocol (SNMP) for information on CSS MIBs.

Table 5-8 describes the fields in the **show mibii** command output.

Table 5-8 Field Descriptions for the show mibii Command

Field	Description
MAC	The interface 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 contains an octet string of zero length.
Administrative	The desired state of the interface (Enabled, Disabled, or Testing). The testing state indicates 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 delivered 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 delivered 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.

Table 5-8 Field Descriptions for the show mibii Command (continued)

Field	Description
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.
Operational	The current operational state of the interface (Up, Down, or Testing). The Testing state indicates 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 contains 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 packets 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 packets 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** - Displays 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* - Displays 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** - Displays the Ethernet errors for all Ethernet interfaces in the CSS and reset the statistics to zero upon retrieval.
- **show ether-errors zero** *interface name* - Displays the Ethernet errors for the specified Ethernet interface in the CSS and resets the statistics to zero upon retrieval. Enter the interface name as a case-sensitive unquoted text string.
- **show ether-errors-32** - Displays the RFC 1398 32-bit statistics, including the **-32** suffix.
- **show ether-errors-32** *interface name* - Displays the RFC 1398 32-bit statistics, including the **-32** suffix. Enter the interface name as a case-sensitive unquoted text string.

Table 5-9 describes the fields in the **show ether-errors** command output.

Table 5-9 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 cyclic redundancy check) 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 frame check sequence (FCS) check.

Table 5-9 Field Descriptions for the show ether-errors Command (continued)

Field	Description
Single Collision	The number of successfully transmitted frames on the interface for transmissions that were inhibited by exactly one collision.
Multiple Collisions	The number of successfully transmitted frames on the interface for transmissions that were 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 failed due to an internal MAC sublayer receive error.
Frame too Long	The number of frames received on the interface that exceeded 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 failed due to an internal MAC sublayer transmit error.
Excessive Collisions	The number of frames for which transmission on the interface failed due to excessive collisions.
Late Collisions	The number of times that a collision is detected on the interface later than 512 bit-times in to the transmission of a packet.

Shutting Down an Interface

Use the **admin-shutdown** command to shut down an interface.

Caution Shutting down an interface terminates all connections to the interface.

To shut down interface e3 on the CSS 11501 enter:

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

Shutting Down All Interfaces

Use the **admin-shutdown** command to shut down all interfaces simultaneously. This command is only available in the SuperUser mode. The **admin-shutdown** command provides a quick way to shut down all physical devices in the CSS except the console and Ethernet management ports.

Caution Shutting down an interface terminates all connections to the interface.

To shut down all interfaces, enter:

```
# admin-shutdown
```

Restarting an Interface

Use the **no admin-shutdown** command to restart the interface. For example to restart interface e3 on the CSS 11501 enter:

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

Note The CSS automatically sends a gratuitous ARP for the IP interface address when you restart the interface. The gratuitous ARP informs all network nodes about ARP mapping. The CSS transmits one ARP request packet and one ARP reply packet for every gratuitous ARP invocation.

Restarting All Interfaces

To restart all interfaces, enter:

```
# no admin-shutdown
```

Note The CSS automatically sends a gratuitous ARP for every configured IP interface address when you restart all interfaces. The gratuitous ARP informs all network nodes about ARP mapping. The CSS transmits one ARP request packet and one ARP reply packet for every gratuitous ARP invocation.

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. Router Discovery Protocol (RDP) settings can also be configured for each circuit VLAN to advertise the CSS to hosts.

This section includes the following topics:

- Entering Circuit Configuration Mode
- Configuring a Circuit IP Interface
- Configuring Router-Discovery Protocol Settings for a Circuit
- Showing Circuits
- Showing IP Interfaces

Entering Circuit Configuration Mode

Use the **circuit** command to enter the circuit configuration mode. Enter the specific VLAN in uppercase letters. Do not include a space between VLAN and the VLAN number. For example:

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

Configuring a Circuit IP Interface

This section includes the following topics:

- Configuring a Circuit IP Address
- Configuring a Circuit-IP Broadcast Address
- Configuring Circuit-IP Redirects
- Configuring Circuit-IP Unreachables
- Configuring Router-Discovery Preference for a Circuit IP Interface
- Enabling and Disabling a Circuit IP

Configuring a Circuit IP Address

Use the **ip address** command to assign an IP address to a circuit. Enter the IP address and a subnet mask in CIDR bit-count notation or a mask in dotted-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 172.16.6.58/8
```

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

```
(config-circuit-ip[VLAN7-172.16.6.58])#
```

Note The CSS automatically sends a gratuitous ARP for the IP interface address when you assign an IP address to a circuit. The gratuitous ARP informs all network nodes about ARP mapping. The CSS transmits one ARP request packet and one ARP reply packet for every gratuitous ARP invocation.

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

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

Configuring a Circuit-IP Broadcast Address

Use the **broadcast** command to change the broadcast address associated with a circuit. 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, IP address 172.16.6.58/24 has a broadcast address of 172.16.6.58/255). This command is available in IP configuration mode.

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

```
(config-circuit-ip[VLAN7-172.16.6.58])# broadcast 0.0.0.0
```

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

```
(config-circuit[VLAN7-172.16.6.58])# no broadcast
```

Configuring Circuit-IP Redirects

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

For example:

```
(config-circuit-ip[VLAN7-172.16.6.58])# redirects
```

To disable the transmission of ICMP redirect messages, enter:

```
(config-circuit-ip[VLAN7-172.16.6.58])# no redirects
```

Configuring Circuit-IP Unreachables

Use the **unreachables** command to enable the transmission of ICMP Destination Unreachable messages. The default state is enabled.

For example:

```
(config-circuit-ip[VLAN7-172.16.6.58])# unreachables
```

To disable the transmission of ICMP Destination Unreachable messages, enter:

```
(config-circuit-ip[VLAN7-172.16.6.58])# no unreachables
```

Configuring Router-Discovery Preference for a Circuit IP Interface

Use the **router-discovery** command to enable router discovery and configure the router discovery preference value for a circuit IP interface. When enabled, router discovery transmits packets with the "all-hosts" multicast address of 244.0.0.1.

Note To enable an interface to transmit packets with the limited broadcast multicast address of 255.255.255.255, use the **router-discovery limited-broadcast** command in circuit mode (see the "Configuring Router-Discovery Limited-Broadcast" section). Router discovery is disabled by default.

Use the **router-discovery preference** command to specify the preference level for the advertised CSS circuit IP address, relative to other devices on the same network. 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, to specify a router discovery preference value of 100, enter:

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

To disable router discovery, enter:

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

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 and Disabling a Circuit IP

Use the **enable** command to enable or disable the IP interface on a circuit. The default is enabled.

For example:

```
(config-circuit-ip[VLAN7-172.16.6.58])# enable
```

To disable the IP interfaces on a circuit, enter:

```
(config-circuit-ip[VLAN7-172.16.6.58])# no enable
```

Configuring Router-Discovery Protocol Settings for a Circuit

The CSS allows you to enable Router Discovery Protocol (RDP) settings and define a router discovery preference for each circuit VLAN. RDP announces the existence of the CSS to hosts by periodically multicasting or broadcasting a router advertisement to each interface.

Use the **circuit** command to enter the circuit configuration mode before configuring RDP for a circuit VLAN.

This section includes the following topics:

- Configuring the Router-Discovery Lifetime
- Configuring Router-Discovery Limited-Broadcast
- Configuring the Router-Discovery Max-Advertisement-Interval
- Configuring the Router-Discovery Min-Advertisement-Interval

Configuring the Router-Discovery Lifetime

Use the **router-discovery lifetime** command to configure the maximum age, in seconds, that hosts remember router advertisements. Enter an integer between 0 and 9000 seconds. The default is three times the **max-advertisement-interval**.

For example:

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

To reset the time to the default of three times the **max-advertisement-interval**, enter:

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

Configuring Router-Discovery Limited-Broadcast

Use the **router-discovery limited-broadcast** command to transmit router discovery packets using the limited broadcast address 255.255.255.255. The default is 224.0.0.1 (the “all-hosts” multicast address).

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 the Router-Discovery Max-Advertisement-Interval

Use the **router-discovery max-advertisement-interval** command to configure the maximum interval timer used for router discovery advertisement from the circuit VLAN. This command defines the maximum interval, in seconds, between sending advertisements. 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 the router discovery maximum advertisement interval to the default of 600, enter:

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

Configuring the Router-Discovery Min-Advertisement-Interval

Use the **router-discovery min-advertisement-interval** command to configure the minimum interval timer used for router discovery advertisement from the circuit VLAN. This command defines the minimum interval, in seconds, between sending advertisements. Enter an integer from 0 to 1800.

The default is 0.75 times the max-advertisement-interval. If this value is greater than 0, it must be less than the value specified using the **router-discovery max-advertisement-interval** command.

For example:

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

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

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

Showing Circuits

Use the **show circuits** command to show circuit information. This command provides the following options:

- **show circuits** - Displays all circuit information for circuits that are currently up
- **show circuits all** - Displays all circuit information regardless of circuit state
- **show circuit name** *circuit name* - Displays 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 5-10 describes the fields in the **show circuits** command output.

Table 5-10 Field Descriptions for the show circuits Command

Field	Description
Circuit Name	The circuit name. The VLAN name appear in uppercase, with no space before the VLAN number.
Circuit State	The state of the circuit. The possible states are as follows: • 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 (Up or Down).

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. For example:

```
# show ip interfaces
```

Table 5-11 describes the fields in the **show ip interfaces** command output.

Table 5-11 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 as follows: • **Active (1)** - The interface is up • **Disabled** - The interface is disabled • **NoCircuit** - 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 the redundancy protocol is running on the interface. The default state is Disabled.
ICMP Redirect	Indicates whether the transmission of Internet Control Message Protocol (ICMP) redirect messages is Enabled or Disabled. The default state is Enabled.
ICMP Unreachable	Indicates whether the transmission of ICMP Destination Unreachable messages is enabled or disabled. The default state is Enabled.
RIP	Indicates whether RIP is Enabled or Disabled.

Configuring RIP for an IP Interface

You can configure Routing Information Protocol (RIP) attributes on each IP interface. To configure 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.

The timers used by RIP in the CSS include the following default values. These RIP timer values are not user-configurable in the CSS.

- Transmit (Tx) time that is a random value between 15 and 45 seconds to avoid router synchronization problems
- Route expiration time of 180 seconds (if the CSS loses the link to the next hop router, the route is immediately removed)
- Hold-down time (the amount of time the CSS transmits with an infinite metric) of 120 seconds

This section includes the following topics:

- Enabling RIP on an IP Interface
- Configuring a RIP Default Route
- Configuring a RIP Receive Version
- Configuring RIP Send Version
- Configuring RIP Packet Logging
- Showing RIP Configurations for IP Addresses

Enabling RIP on an IP Interface

Use the **rip** command to start running RIP on an IP interface. For example:

```
(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 a RIP Default Route

Use the **rip default-route** command to advertise a default route on an IP interface with a specific metric. 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 a RIP Receive Version

Use the **rip receive** command to specify the RIP version that the interface receives. The options for this command are as follows:

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

For example:

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

Configuring RIP Send Version

Use the **rip send** command to specify the RIP version that the interface transmits. The options for this command are as follows:

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

For example:

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

Configuring RIP Packet Logging

Use the **rip log** command to enable the CSS to log received or transmitted RIP packets on the interface. Use the **no** form of this command to disable logging (default setting).

The options for this command are as follows:

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

For example:

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

Showing RIP Configurations for IP Addresses

Use the **show rip** command to show a RIP configuration for one IP address or all IP addresses configured in the CSS. The options for this command are as follows:

- **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 5-12 describes the fields in the **show rip** command output.

Table 5-12 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 values are as follows: • **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 as follows: • **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 5-13 describes the fields in the **show rip globals** command output.

Table 5-13 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 5-14 describes the fields in the **show rip statistics** command output.

Table 5-14 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

Configuring the Switched Port Analyzer Feature

Configure the switched port analyzer (SPAN) feature on your CSS to mirror (copy) traffic passing through one CSS port (Fast Ethernet or Gigabit Ethernet) to another designated port of the same type and on the same CSS module for analysis. You can use SPAN for network troubleshooting or tuning using a network analyzer. SPAN is sometimes referred to as *port mirroring* or *port monitoring*.

A SPAN session is the association of a destination port with a source port on the same CSS module. The port that is monitored is called the source SPAN (SSPAN) port. An SSPAN port consists of two components:

- Ingress path - Network traffic entering the CSS. The CSS copies to the monitoring port packets that the SSPAN port receives (SSPAN Rx) from the network.
- Egress path - Network traffic leaving the CSS. The CSS copies to the monitoring port packets that the SSPAN port transmits (SSPAN Tx) to the network.

SPAN can monitor the ingress path, the egress path, or both. You can configure only one SSPAN port in a CSS chassis.

The port that monitors the SSPAN port is called the destination SPAN (DSPAN) port. You can configure only one DSPAN port in a CSS chassis and it must have the following characteristics:

- Same speed as the SSPAN port
- Same media type as the SSPAN port
- Local (physically resides on the same CSS module)

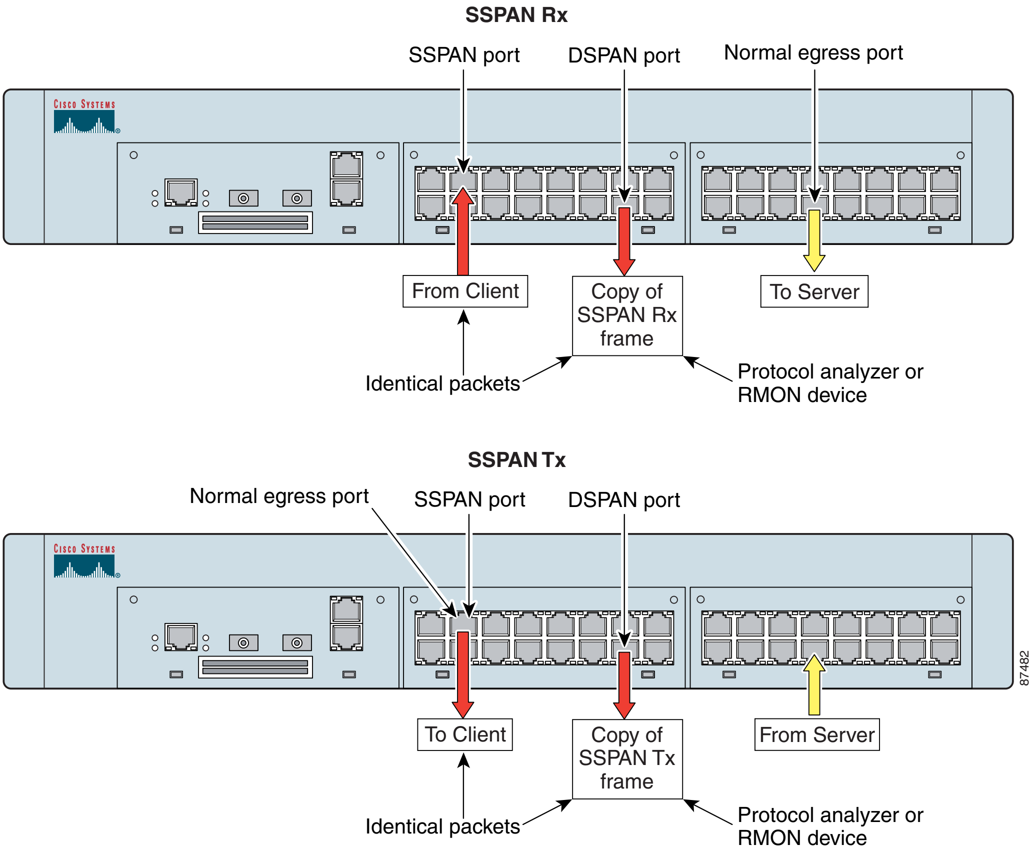
Once you configure a port as a DSPAN port, the CSS removes it from all VLANs and ignores ingress traffic on that port. In addition, the DSPAN port does not participate in STP or routing protocols such as RIP and OSPF.

Traffic copied to the DSPAN port is typically forwarded to a network analyzer, protocol analyzer, or an RMON probe. SPAN allows you to monitor CSS ports without:

- Disconnecting cables
- Requiring multiple analyzers or probes
- Needing hubs or switches

Figure 5-3 shows an example of SPAN connectivity with a protocol analyzer connected to port 2/13 on a CSS. In this example, the CSS copies all packets received or transmitted on Fast Ethernet (FE) port 2/4 (SSPAN port) to FE port 2/13 (DSPAN port). The analyzer connected to DSPAN port 2/13 receives all network traffic that the SSPAN port receives or transmits.

Figure 5-3 Example of SPAN Connectivity



This section describes how to configure SPAN on a CSS. It includes the following topics:

- Configuring SPAN on a CSS
- Verifying the SPAN Configuration on a CSS

Configuring SPAN on a CSS

To configure SPAN on a CSS, use the **setspan** command. This command instructs the CSS to monitor all incoming and/or outgoing traffic on a specified SSPAN port by copying the packets to a specified DSPAN port on the same module in the CSS. This feature is disabled by default.

The syntax of this global configuration mode command is:

```
setspan src_port number dest_port number
    copyBoth|copyTxOnly|copyRxOnly
```

The options and variables for this command are as follows:

- **src_port** *number* - Source port keyword and number of the SSPAN port (in slot/port format) that you want to monitor. The CSS copies all packets that are received or transmitted on this port to the DSPAN port.
- **dest_port** *number* - Destination port keyword and number of the DSPAN port (in slot/port format) where you want to connect the network analyzer, protocol analyzer, or RMON probe. The CSS copies the packets that flow through the SSPAN port to the DSPAN port that you specify. The DSPAN port must reside on the same module as the SSPAN port.

  Note Once you configure a port as a DSPAN port, the CSS removes it from all VLANs and ignores ingress traffic on that port. In addition, the DSPAN port does not participate in spanning tree protocol (STP) or routing protocols such as RIP and OSPF.

- **copyBoth** - CSS copies to the DSPAN port packets that the SSPAN port transmits to the network (egress traffic) and packets that the SSPAN port receives from the network (ingress traffic).

Note If the combined traffic bandwidth of the ingress and egress traffic of the SSPAN port exceeds the bandwidth of the DSPAN port, the DSPAN port may become oversubscribed.

- **copyTxOnly** - CSS copies to the DSPAN port only those packets that the SSPAN port transmits to the network (egress traffic).
- **copyRxOnly** - CSS copies to the DSPAN port only those packets that the SSPAN port receives from the network (ingress traffic).

For example, to copy all received and transmitted packets on SSPAN port 3 of the I/O module in slot 3 to DSPAN port 12 on the same module, enter:

```
(config)# setspan src_port 3/3 dest_port 3/12 copyBoth
```

To return the SPAN feature to its default state of disabled, use the **no setspan** command. For example, to disable SPAN on the source and destination ports on CSS module 3 in the example above, enter:

```
(config)# no setspan src_port 3/3 dest_port 3/12
```

Verifying the SPAN Configuration on a CSS

To verify the SPAN configuration on a CSS, use the **show setspan** command. Table 5-15 describes the fields in the **show setspan** command output.

Table 5-15 Field Descriptions for the show setspan Command

Field	Description
SPAN Configuration	
Source	Number of the SSPAN port whose traffic you want to monitor.
Destination	Number of the DSPAN port to which the CSS copies the packets flowing through the SSPAN port. Connect the network analyzer or RMON probe to this port.

Table 5-15 Field Descriptions for the show setspan Command (continued)

Field	Description
Direction	Direction of the traffic that you want to monitor at the source port. The direction can be one of the following: • **copyBoth** - The CSS copies packets that are transmitted and received by the SSPAN port to the DSPAN port. • **copyTxOnly** - The CSS copies only packets transmitted (egress traffic) by the SSPAN port to the DSPAN port. • **copyRxOnly** - The CSS copies only packets received (ingress traffic) by the SSPAN port to the DSPAN port.

Where to Go Next

Chapter 6, Configuring CSS Network Protocols describes how to configure Domain Name Service (DNS), Address Resolution Protocol (ARP), Routing Information Protocol (RIP), Internet Protocol (IP), and spanning-tree bridging, and Dynamic Host Configuration Protocol (DHCP).

Configuring CSS Network Protocols

This chapter describes how to configure Domain Name Service (DNS), Address Resolution Protocol (ARP), Routing Information Protocol (RIP), Internet Protocol (IP) routing, spanning-tree bridging, and Dynamic Host Configuration Protocol (DHCP). Information in this chapter applies to all CSS models, except where noted.

This chapter includes the following major sections:

- Configuring the Domain Name Service
- Configuring the Address Resolution Protocol
- Configuring Routing Information Protocol
- Configuring the Internet Protocol
- Configuring the Cisco Discovery Protocol
- Configuring Spanning-Tree Bridging for the CSS
- Configuring the DHCP Relay Agent

Configuring the Domain Name Service

Use the **dns** command to enter commands that control Domain Name Service (DNS), the facility that translates host names such as myhost.mydomain.com to IP addresses such as 192.168.11.1.

This section includes the following topics:

- Specifying a Primary DNS Server
- Using DNS Resolve
- Specifying a Secondary DNS Server
- Specifying a DNS Suffix
- Specifying UDP Traffic on the DNS Server Port

Use the **show running-config global** command to display DNS configurations (refer to Chapter 3, Managing the CSS Software).

Specifying a Primary DNS Server

Use the **dns primary** command to specify the primary DNS server. Enter the IP address in dotted-decimal notation (for example, 192.168.11.1) of the DNS server you want to specify as the primary DNS server.

For example:

```
(config)# dns primary 192.168.11.1
```

To remove the primary DNS server, enter:

```
(config)# no dns primary
```

Using DNS Resolve

Use the **dns resolve** command to resolve a host name by querying the DNS server. Enter the host name you want to resolve in mnemonic host-name format (for example, myhost.mydomain.com).

For example:

```
(config)# dns resolve fred.arrowpoint.com
```

Specifying a Secondary DNS Server

When a primary DNS server fails, the CSS uses the secondary DNS server to resolve host names to IP addresses. Use the **dns secondary** command to specify a secondary DNS server. Enter the IP address of the secondary DNS server in dotted-decimal notation (for example, 192.168.11.1).

```
(config)# dns secondary 192.168.3.6
```

You can specify a maximum of two secondary servers. To specify each additional server, repeat the **dns secondary** command. The order in which you enter the IP addresses is the order in which they are used when the primary DNS server fails.

To remove a secondary DNS server, specify the **no** version of the command followed by the IP address of the DNS server you wish to remove. For example:

```
(config)# no dns secondary 192.168.3.6
```

Specifying a DNS Suffix

Use the **dns suffix** command to specify the default suffix to use when querying the DNS facility. Enter the default suffix as an unquoted text string with no spaces and a maximum of 64 characters.

For example:

```
(config)# dns suffix arrowpoint.com
```

To remove the default DNS suffix, enter:

```
(config)# no dns suffix
```

Specifying UDP Traffic on the DNS Server Port

For DNS UDP traffic on port 53, use the **dnsflow** command to determine whether the CSS uses flow control blocks (FCBs) for DNS requests and responses. This command provides the following options:

- **enable** (default) - Causes the CSS to set up flows using FCBs for DNS requests and responses. Because UDP traffic is connectionless, the DNS flows remain active until the flow manager reclaims the flow resources.
- **disable** - Causes the CSS to not use FCBs for the DNS requests and responses. Use this setting for sites with heavy DNS traffic or sites where the DNS clients use a source and destination port of 53.

For example:

```
(config)# dnsflow disable
```

Configuring the Address Resolution Protocol

Use the **arp** command to statically configure the IP to Media Access Control (MAC) translations necessary for the CSS to send data to network nodes. You can configure static ARP mapping for any of the CSS Ethernet interface ports.

This section includes the following topics:

Configuring ARP

Use the **arp** command to define a static ARP mapping. The syntax for this global configuration mode command is:

arp *ip_or_host mac_address interface* {*vlan*}

The variables and options are as follows:

- *ip_or_host* - The IP address of the system for static mapping. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1) or in mnemonic host-name format (for example, myhost.mydomain.com).
- *mac_address* - The MAC address of the system mapped to the IP address. Enter the MAC address in hyphenated-hexadecimal notation (for example, 00-60-97-d5-26-ab).
- *interface* - The CSS Ethernet interface port that you want to configure. For a CSS 11501, enter the interface name in *interface port* format (for example, e2). For a CSS 11503 or CSS 11506, the interface format is *slot/port* (for example, 3/1).
- *vlan* - The number of the VLAN configured in a trunked interface on which the ARP address is configured (assuming trunking is enabled for the CSS Gigabit Interface port). Enter an integer from 1 to 4094 as the VLAN number.

For example:

```
(config)# arp 192.168.11.1 00-60-97-d5-26-ab e2
```

To remove a static mapping address, use the **no arp** command. For example:

```
(config)# no arp 192.168.11.1
```

Note The CSS discards ARP requests from hosts that are not on the same network as the CSS circuit IP address. Thus, if a CSS and a host are within the same VLAN but configured for different IP networks, the CSS does not respond to ARP requests from the host.

Configuring ARP Timeout

Use the **arp timeout** command to set the time, in seconds, to hold an ARP resolution result. When you change the timeout value, this value affects only new ARP entries. All previous ARP entries retain the old timeout value. To remove all entries with the old timeout value, enter the **clear arp cache** command.

The timeout value is the number of seconds the CSS holds an ARP resolution result. To set a timeout value, enter an integer from 60 to 86400 (24 hours) seconds. The default is 14400 seconds (4 hours). If you do not want the ARP entries to time out, enter **none** or **86401**.

For example:

```
(config)# arp timeout 120
```

To restore the default timeout value of 14400 seconds, enter:

```
(config)# no arp timeout
```

Configuring ARP Wait

Use the **arp wait** command to set the time, in seconds, to wait for an ARP resolution. The wait time is the number of seconds the CSS waits for an ARP resolution in response to an ARP request to the network. Enter an integer from 5 to 30 seconds. The default is 5.

For example:

```
(config)# arp wait 15
```

To restore the default wait time of 5 seconds, enter:

```
(config)# no arp wait
```

Updating ARP Parameters

Use the **update arp** command to update the file containing hosts reachable through ARP. This command is available only in SuperUser mode. For example:

```
# update arp file
```

Clearing ARP Parameters

The CSS enables you to clear ARP parameters for the ARP file or ARP cache. To clear the file that contains known hosts reachable through ARP, use the **clear arp file** command. For example:

```
# clear arp file
```

Use the **clear arp cache** command to delete dynamic entries from the ARP cache. To specify an address for the single ARP entry you want to remove from the ARP cache, use the **clear arp cache** *ip_or_host* command. Enter the address in either dotted-decimal IP notation (for example, 192.168.11.1) or mnemonic host-name format (for example, myhost.mydomain.com).

For example:

```
# clear arp cache 192.168.11.1
```

Showing ARP Information

Use the **show arp** command to display ARP information. To show static ARP mapping when you use the **show arp** command, the IP route must exist in the routing table.

The syntax for this global configuration mode command is:

show arp {**config**|**file**|**management-port**|**summary**|*ip_or_host*}

The syntax and options for the command are as follows:

- **show arp** - Displays the complete ARP resolution table with IP addresses, MAC addresses, and resolution type, excluding entries from the CSS Ethernet management port.
- **config** - Displays ARP global configuration parameters. The screen displays the response timeout and the flush timeout, in seconds.
- **file** - Displays the hosts that are reachable using ARP. The screen displays the IP addresses of the host systems.
- **management-port** - Displays the ARP entries from the CSS Ethernet management port. The ARP resolution table displayed through the **show arp** command displays these entries.

Note The CSS Ethernet management port IP address appears as an entry in the Management Port ARP cache. This is normal CSS behavior.

- **summary** - Displays the total number of static entries, total number of dynamic entries, and total number of entries in the ARP resolution table, excluding the entries from the CSS management port.
- *ip_or host* - The IP address for the system to display its resolution. Enter the address in dotted-decimal format (for example, 192.168.11.1) or mnemonic host-name format (for example, myname.mydomain.com). You cannot enter an ARP entry derived from the CSS Ethernet management port.

For example, to display the complete ARP resolution table, enter:

```
# show arp
```

Table 6-1 describes the fields in the **show arp** command output.

Table 6-1 Field Descriptions for the show arp Command

Field	Description
IP Address	The IP address of the system for ARP mapping.
MAC Address	The MAC address of the system mapped to the IP address.
Type	The resolution type for the entry: Dynamic or Static. The Dynamic resolution type indicates that the entry was discovered through the ARP protocol. The Static resolution type indicates that the entry is from a static configuration.
Port	The CSS interface configured as the egress logical port.

To display a summary of entries in the ARP resolution table, enter:

```
# show arp summary
```

Table 6-2 describes the fields in the **show arp summary** command output.

Table 6-2 Field Descriptions for the show arp summary Command

Field	Description
Static Entry	The total number of static map entries in the ARP resolution table (from a static configuration).
Dynamic Entry	The total number of dynamic map entries in the ARP resolution table (entries discovered through the ARP protocol).
Total Entry	The total number of static and dynamic entries in the ARP resolution table.

To display the global ARP configuration, enter:

```
# show arp config
```

Table 6-3 describes the fields in the **show arp config** command output.

Table 6-3 Field Descriptions for the show arp config Command

Field	Description
ARP Response Timeout	The time, in seconds, to wait for an ARP resolution response before discarding the packet waiting to be forwarded to an address. The time can be from 5 to 30 seconds. The default is 5 seconds.
ARP Flush Timeout	The time, in seconds, to hold an ARP resolution result in the ARP cache. The timeout period can be from 60 to 86400 seconds (24 hours). The default is 14400 seconds (4 hours). An entry of none or 86401 indicates the ARP entries will not timeout.

To display the host IP addresses entered at initialization or boot time through ARP, enter:

```
# show arp file
```

To display the ARP entries from the CSS management port, enter:

```
# show arp management-port
```

Table 6-4 describes the fields in the **show arp management-port** command output.

Table 6-4 Field Descriptions for the show arp management-port Command

Field	Description
IP Address	The IP address of the system for ARP mapping.
MAC Address	The MAC address of the system mapped to the IP address.
Port	The CSS Ethernet management port.

To display the resolution for a host IP address, enter:

```
# show arp 192.50.1.6
```

To display the host IP addresses entered at initialization or boot time through ARP, enter:

```
# show arp file
```

Configuring Routing Information Protocol

The CSS enables you to configure global Routing Information Protocol (RIP) attributes used to advertise routes on the CSS. By default, RIP advertises RIP routes and local routes for interfaces running RIP. The **rip** command advertises other routes.

The timers used by RIP in the CSS include the following default values. These RIP timer values are not user-configurable in the CSS.

- Transmit (Tx) time that is a random value between 15 and 45 seconds (it avoids router synchronization problems
- Route expiration time of 180 seconds (if the CSS loses the link to the next hop router, the route is immediately removed).
- Hold-down time (the amount of time the CSS transmits with an infinite metric) of 120 seconds.

This section includes the following topics:

- Configuring RIP Advertise
- Configuring RIP Redistribute
- Configuring Equal-Cost RIP Routes
- Showing RIP Configurations

Note If you prefer OSPF instead of RIP on the CSS, refer to Chapter 7, Configuring Open Shortest Path First (OSPF) for information on configuring OSPF.

Configuring RIP Advertise

Use the **rip advertise** command to advertise a route through RIP on the CSS. The syntax for this command is:

rip advertise *ip_address subnet_mask* {*metric*}

The variables for this command are as follows:

- *ip_address* - The IP address for the route prefix. Enter an IP address in dotted-decimal notation (for example, 192.168.1.0).
- *subnet_mask* - The IP prefix length in CIDR bitcount notation (for example, /24) or in dotted-decimal notation (for example, 255.255.255.0).
- *metric* - (Optional) Metric to use when advertising this route. Enter a number from 1 to 15. The default is 1.

For example:

```
(config)# rip advertise 192.168.1.0/24 9
```

Note The network does not have to be present in the routing table to be advertised. The **SNTP ip advertise** command is intended for advertising VIP addresses.

To stop advertising a route through RIP on the CSS, enter:

```
(config)# no rip advertise 192.168.1.0/24
```

Configuring RIP Redistribute

Use the **rip redistribute** command to advertise routes from other protocols through RIP. By default, RIP advertises RIP routes and local routes for interfaces running RIP. This command instructs RIP to advertise other routes, such as firewall routes, OSPF routes, and so on.

The syntax for this command is

rip redistribute [**firewall|local|ospf|static**] {*metric*}

The options and variables for this command are as follows:

- **firewall** - Advertises firewall routes through RIP.
- **local** - Advertises local routes (interfaces *not* running RIP).
- **static** - Advertises static routes configured for the Ethernet interface ports.
- **ospf** - Advertises OSPF routes through RIP.
- *metric* - (Optional) Metric to use when advertising this route. Enter a number from 1 to 15. The default is 1.

For example:

```
(config)# rip redistribute static 3
```

To stop advertising routes from other protocols through RIP, use either the **local, static**, or **firewall** option.

The following commands stop advertising static routes:

```
(config)# no rip redistribute firewall
(config)# no rip redistribute local
(config)# no rip redistribute static
(config)# no rip redistribute ospf
```

Configuring Equal-Cost RIP Routes

Use the **rip equal-cost** command to set the maximum number of routes that RIP can insert into the routing table. Enter a number from 1 to 15. The default is 1. For example:

```
(config)# rip equal-cost 4
```

To reset the number of routes to the default value of 1, enter:

```
(config)# no rip equal-cost
```

Showing RIP Configurations

Use the **show rip** command to show a RIP configuration for one IP address or all IP addresses configured in the CSS. This command provides the following options and variables:

- **show rip** - Displays RIP configurations for all interfaces
- **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 6-5 describes the fields in the **show rip** command output.

Table 6-5 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 as follows: • **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 as follows: • **both** - Receive both version 1 and version 2 (default) • **none** - Receive no RIP packets • **Ripv1** - Receive RIP version 1 packets only • **Ripv2** - Receive RIP version 2 packets only
Default Metric	The default metric used for advertising the RIP interface.

Table 6-5 Field Descriptions for the show rip Command (continued)

Field	Description
Tx Log	The setting for logging RIP packet transmissions (enabled or disabled). The default setting is disabled.
Rx Log	The setting for logging RIP packets received (enabled or disabled). The default setting is disabled.

To display global RIP statistics, enter:

```
# show rip globals
```

Table 6-6 describes the fields in the **show rip globals** command output.

Table 6-6 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 6-7 describes the fields in the **show rip statistics** command output.

Table 6-7 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

Configuring the Internet Protocol

Use the **ip** command to specify Internet Protocol (IP) configuration commands for the CSS. This command is available in global configuration mode.

This section includes the following topics:

- Configuring an IP Route
- Disabling an Implicit Service for the Static Route Next Hop
- Configuring an IP Source Route
- Configuring the IP Record Route
- Configuring Box-to-Box Redundancy
- Configuring IP Equal-Cost Multipath
- Forwarding IP Subnet Broadcast Addressed Frames
- Configuring IP Unconditional Bridging
- Configuring IP Opportunistic Layer 3 Forwarding
- Showing IP Configuration Information

Configuring an IP Route

A static route consists of a destination network address and mask, as well as the next hop to reach the destination. You can also specify a default static route (using 0.0.0.0 as the destination network address and a valid next hop address) to direct frames for which no other destination is listed in the routing table. Default static routes are useful for forwarding otherwise unrouteable packets by the CSS.

When you configure a static route, the CSS creates an internal service that periodically polls the configured next hop address with an ICMP echo (or ping) keepalive. The internal service is called an implicit service. If the router fails, the CSS removes any entries from the routing table that point to the failed router and stops sending network traffic to the failed router. When the router recovers, the CSS:

- Becomes aware of the router
- Reenters applicable routes into the routing table

The implicit service does not determine if the default or static route appears in the routing table. This decision is based on the CSS having a viable ARP entry for the next hop router IP address so the CSS can forward traffic to that destination. The CSS uses the ICMP keepalive as a means to ensure the next hop router MAC address is available and current. However, in certain situations, the next hop router may block ICMP message transmitted by the CSS, which results in a failed ICMP keepalive (the ICMP keepalive is in the Down state). As long as the CSS has the ARP entry of the next hop router the static route is still placed in the routing table.

Note The CSS allows you to disable the internal ICMP keepalive through the **ip-no-implicit service** command. In this case, if the MAC address for the next hop is not known to the CSS the address will not appear in the routing table.

Use the **ip route** command to configure an IP route. You can configure a static route, a default static IP route, a blackhole route (where the CSS drops any packets addressed to the route), or a firewall IP route. Each **ip route** command requires one of the following:

- An IP address and a subnet mask prefix; for example, 192.168.1.0 /24
- An IP address and a subnet mask; for example, 192.168.1.0 255.255.255.0

The syntax for this global configuration command is:

ip route *ip_address subnet_mask*[**blackhole**|*ip_address2*{*distance*|
originated-packets}|**firewall** *index* {*distance*}]

The syntax and options for the command are as follows:

- *ip_address* - The destination network address. Enter the IP address in dotted-decimal notation (for example, 192.168.11.1).
- *subnet_mask* - The IP subnet mask. Enter the mask in either:
  - CIDR bitcount notation (for example, /24).
  - Dotted-decimal notation (for example, 255.255.255.0).
- **blackhole** - Instructs the CSS to drop any packets addressed to the destination.
- *ip_address2* - The next hop address for the route. Enter the IP address in dotted-decimal notation (for example, 192.168.11.1).

- *distance* - (Optional) The administrative distance. Enter an integer from 1 to 254. A smaller number is preferable. The default value is 1.
- **originated-packets** - Specifies that the route is used only by packets created using flows or sessions going to and from the CSS (for example, a Telnet session to the CSS). The route is not used by flows or sessions that go through the CSS (for example, between an attached server and a remote client).

Note A ping response and an SNMP responses do not use the originated-packets route. A ping *request* sent from the CSS uses the originated-packets route. A ping *response* sent from the CSS does not use the originated-packets route.

- **firewall** - Configures a firewall route. The **firewall** option instructs the CSS to use firewall load balancing for this route. You can optionally set the administrative distance.

Note The CLI prevents you from configuring IP static routes with identical destinations *and* identical administrative costs, for IP static routes that are firewall routes and IP static routes that are not firewall routes.

- *index* - An existing index number for the firewall route. For information on configuring a firewall index, see the **ip firewall** command (refer to the *Cisco Content Services Switch Advanced Configuration Guide*).

For example, to configure a static IP route to destination network address *192.168.0.0 /16* and a next hop address of *192.167.1.1*, enter:

```
(config)# ip route 192.168.0.0 /16 192.167.1.1
```

For example, to configure a default IP route using a destination address of *0.0.0.0 /0* and a next hop address of *192.167.1.1*, enter:

```
(config)# ip route 0.0.0.0 /0 192.167.1.1
```

For example, to configure a blackhole route, enter:

```
(config)# ip route 192.168.1.0 /24 blackhole
```

For example, to configure a firewall IP route with an index number of *3* and an administrative distance of *2*, enter:

```
(config)# ip route 192.168.1.0 /24 firewall 3 2
```

To remove a static route, enter:

```
(config)# no ip route 0.0.0.0 /0 10.0.1.1
```

To disable the dropping of packets to a blackhole route, enter:

```
(config)# no ip route 192.168.1.0 /24 blackhole
```

To remove a firewall route, enter:

```
(config)# no ip route 192.168.1.0 /24 firewall 3
```

Disabling an Implicit Service for the Static Route Next Hop

Use the **ip no-implicit-service** command when you do not want the CSS to start an implicit service for the next hop of a static route. By default, the CSS establishes an implicit (or internal) service for the gateway address when a static route is defined. The **ip no-implicit-service** command specifies that no implicit service is established to the next hop of the static route, which disables the internal service ICMP keepalive. In this case, if the ARP address for the next hop is not known to the CSS, the address will not appear in the routing table.

The purpose of the implicit service to the next hop of a static route is to monitor the availability of the next hop to forward data traffic. When the **ip no-implicit-service** command is in effect, traffic is forwarded to the next hop even when the next hop is unavailable. Because of the possibility of data being lost if the next hop becomes unavailable, use of the **ip no-implicit-service** command is strongly discouraged.

Note Static routes can sometimes appear in the CSS routing table even when you have an implicit service for the next hop address (the default setting) and the internal keepalive is down. When the CSS detects the ARP mapping for the next hop in the static route, the CSS continues to list that route in the routing table regardless of the state of the ICMP service keepalive (Down or Up).

When you implement the **ip no-implicit-service** global configuration command, this action does not affect previously configured static routes. The **ip no-implicit-service** command affects only those static routes added after you enable the command. We recommend you reboot the CSS after you modify the configuration to ensure all static routes are the same, which is useful for network monitoring and troubleshooting. If you wish to stop the implicit service for a previously configured static route, then you must delete and reconfigure the static route.

For example:

```
(config)# ip no-implicit-service
```

To reset the default setting, enter:

```
(config)# no ip no-implicit-service
```

Configuring an IP Source Route

Use the **ip source-route** command to enable the CSS to process frames with information that overrides the default routing. For example:

```
(config)# ip source-route
```

Caution Enabling the **ip source-route** command may pose a major security risk to your network. The IP source route specifies information that overrides the default routing a packet would normally take. The packet could then bypass a firewall. If this poses a problem, avoid using the **ip source-route** command.

The CSS does not load balance TCP or UDP packets with IP options that are destined to a VIP address. These packet types are dropped and the CSS returns an ICMP destination/port unreachable error. This behavior exists regardless of the state (enabled or disabled) of the **ip source-route** and **ip record-route** commands.

The CSS, however, does respond to ICMP packets that are destined to a VIP address. The CSS also responds to TCP or UDP packets that include IP options that are destined to a local circuit address, or require that a routing decision be made.

To disable the processing of frames with the IP source-route option (the default behavior), enter:

```
(config)# no ip source-route
```

Configuring the IP Record Route

Use the **ip record-route** command to enable the CSS to process frames with the IP address of each router along a path. For example:

```
(config)# ip record-route
```

Caution Enabling the **ip record-route** command could pose security risks to your network. The **ip record-route** command inserts the IP address of each router along a path into the IP header.

The CSS does not load balance TCP or UDP packets with IP options that are destined to a VIP address. These packet types are dropped and the CSS returns an ICMP destination/port unreachable error. This behavior exists regardless of the state (enabled or disabled) of the **ip record-route** and **ip source-route** commands.

The CSS, however, does respond to ICMP packets that are destined to a VIP address. The CSS also responds to TCP or UDP packets that include IP options that are destined to a local circuit address, or require that a routing decision be made.

To disable the processing of frames with the record-route option (the default behavior), enter:

```
(config)# no ip record-route
```

Configuring Box-to-Box Redundancy

Use the **ip redundancy** command to enable box-to-box redundancy. Box-to-box redundancy provides chassis-level redundancy between two identically configured CSSs. Refer to the *Cisco Content Services Switch Advanced Configuration Guide* for information about configuring box-to-box redundancy.

The CSS does not support simultaneous box-to-box redundancy and VIP or interface redundancy configurations.

For example:

```
(config)# ip redundancy
```

To disable box-to-box redundancy, enter:

```
(config)# no ip redundancy
```

Configuring IP Equal-Cost Multipath

Use the **ip ecmp** command to set the equal-cost multipath (ECMP) selection algorithm and the preferred reverse egress path. The CSS supports a maximum of 15 ECMP paths.

The syntax for this global configuration command is:

ip ecmp [**address**|**no-prefer-ingress**|**roundrobin**]

The options for this global configuration mode command are as follows:

- **address** - Choose among alternate paths based on IP addresses. For example:

  ```
  (config)# ip ecmp address
  ```

- **no-prefer-ingress** - Do not prefer the ingress path of a flow for its reverse egress path. By default, the ingress path for a flow is the preferred egress path. This means that the preferred interface over which to reply to a client is the interface on which the CSS originally received the request from the client. For example:

  ```
  (config)# ip ecmp no-prefer-ingress
  ```

  To reset the ingress path of a flow for its preferred reverse egress path, enter:

  ```
  (config)# no ip ecmp no-prefer-ingress
  ```

- **roundrobin** - Alternate between equal paths in roundrobin fashion. For example:

  ```
  (config)# ip ecmp roundrobin
  ```

Note The CSS applies the ECMP selection algorithm for non-TCP/UDP packets (for example, ICMP) on a packet-by-packet basis. Multipath selection for TCP and UDP is performed on a per-flow basis, and all packets for a particular flow take the same path.

Forwarding IP Subnet Broadcast Addressed Frames

Use the **ip subnet-broadcast** command to enable the CSS to forward subnet broadcast addressed frames.

For example:

```
(config)# ip subnet-broadcast
```

To disable forwarding of subnet broadcast addressed frames (the default behavior), enter:

```
(config)# no ip subnet-broadcast
```

Caution Enabling the CSS to forward the subnet broadcast can make the subnet susceptible to "smurf" attacks; an attacker sends an ICMP echo request frame using a subnet broadcast address as a destination and a forged address as the source.

If a "smurf" attack is successful, all the destination subnet hosts reply to the echo and flood the path back to the source. By disabling subnet broadcast forwarding, the original echo never reaches the hosts.

Configuring IP Unconditional Bridging

By default, the routing table lookup of a destination path by the CSS on received packets overrides bridging decisions to be made for those packets. If the routing table specifies that the CSS use a different physical Ethernet port than what is specified for port bridging, the CSS ignores the bridging decision. If you have a network that you want to bridge through the CSS to an upstream router, you may want to force the CSS to make a bridging decision on the received packets instead of making a routing table decision.

Use the **ip uncond-bridging** global configuration command to always make a bridging decision on the received packets. With this command, the bridging decision always takes precedence over a routing table decision.

For example:

```
(config)# ip uncond-bridging
```

To restore the default behavior of the CSS, enter:

```
(config)# no ip uncond-bridging
```

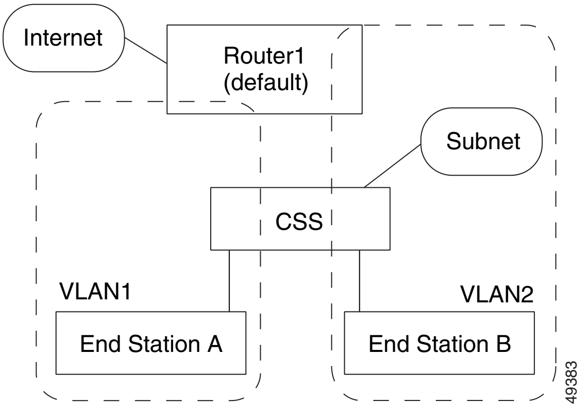
Configuring IP Opportunistic Layer 3 Forwarding

The CSS opportunistic Layer 3 forwarding feature allows the CSS to reduce the number of network device hops for certain packets or flows. The CSS forwards packets at Layer 3 if the destination MAC address in the Ethernet header is the CSS MAC address. Use the **ip opportunistic** command to enable opportunistic Layer 3 forwarding and allow the CSS to make Layer 3 forwarding decisions even if the Layer 2 packet destination MAC address does not belong to the CSS.

For example, Figure 6-1 shows a CSS connected to VLAN1 and VLAN2. Each VLAN has an end station and an uplink to Router1. End stations A and B both point to Router1 as their default router. When End Station A transmits a packet to End Station B, it uses its default route to Router1. The packet contains Router1's destination MAC address. A traditional Layer 2 device forwards the packet to Router1, and Router1 forwards the packet to End Station B on VLAN2.

Using opportunistic Layer 3 forwarding, the CSS inspects the IP packet header to determine the destination IP address. Instead of forwarding the packet to Router1, the CSS forwards the packet directly to End Station B. Because the CSS handles the packet only once, the router and uplink are not used and network resources are conserved.

Figure 6-1 Example of Opportunistic Layer 3 Forwarding



The options for this global configuration mode command are as follows:

- **local (default)** - Applies opportunistic Layer 3 forwarding if the destination IP address belongs to a node that resides on one of the subnets directly attached to the CSS *and* the CSS is aware of an ARP resolution for that node. Because the local option is the default, use the **no ip opportunistic** command to reconfigure IP opportunistic Layer 3 forwarding to the local setting.
- **all** - Applies opportunistic Layer 3 forwarding if the destination IP address matches any entry in the CSS routing table. We do not recommend this option if the topology includes multiple routers and the CSS does not know all of the routes the routers are aware of.
- **disabled** - The CSS does not perform opportunistic Layer 3 forwarding. Regular Layer 3 forwarding is performed only for packets that contain the CSS destination MAC address.

For example, to configure IP opportunistic Layer 3 forwarding to **all**, enter:

```
(config)# ip opportunistic all
```

To reconfigure IP opportunistic Layer 3 forwarding to the default of **local** enter:

```
(config)# no ip opportunistic
```

When you configure **ip opportunistic all**, you can use the **ip route originated-packets** command (see the "Configuring an IP Route" section) to configure routes that the CSS uses to reach devices, but does not use as opportunistic routes for forwarding traffic. Routes created using the **ip route originated-packets** command apply only to packets that originate on the CSS. Packets and flows forwarded by the CSS do not use these routes.

For example:

```
(config)# ip route 0.0.0.0 /0 192.168.1.7 originated-packets
```

Showing IP Configuration Information

Use the **show ip** command to display IP information for the CSS. This section includes the following topics:

- Showing IP Global Configuration Parameters
- Showing IP Interface Information
- Showing IP Routing Information
- Showing IP Statistics
- Showing a Summary of IP Global Statistics

Showing IP Global Configuration Parameters

Use the **show ip config** command to display IP global configuration parameters. These parameters show the state (enabled or disabled) of the source route option, forward IP broadcasts, record-route option, and IP route change logging. The **show ip config** command also shows the value for the orphaned route timer.

Table 6-8 describes the fields in the **show ip config** output.

Table 6-8 Field Descriptions for the show ip config Command

Field	Description
Source Route Option	Indicates whether processing of source-routed frames is enabled or disabled.
Forward IP Broadcasts	Indicates whether forwarding IP broadcasts is enabled or disabled.
Orphaned Route Timer	The setting for the orphaned route timer.
Record Route Option	Indicates whether processing with the record-route option is enabled or disabled.

Table 6-8 Field Descriptions for the show ip config Command (continued)

Field	Description
Multiple Equal Cost Path Algorithm	The setting for the equal-cost multipath selection algorithm. The possible settings are as follows: • **Address** - Choose among alternate paths based on IP addresses • **Roundrobin** - Alternate between equal paths in roundrobin fashion
IP Route Change Logging	Indicates whether logging IP route changes is enabled or disabled.

Showing IP Interface Information

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 6-9 describes the fields in the **show ip interfaces** command output.

Table 6-9 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 as follows: • **Active (1)** - Interface is up • **Disabled (2)** - Interface is disabled • **NoCircuit (3)** - Interface is waiting for an underlying circuit
IP Address	The IP address assigned to the circuit.
Network Mask	The network mask of the circuit.

Table 6-9 Field Descriptions for the show ip interfaces Command (continued)

Field	Description
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 the redundancy protocol is running on the interface. The default state is Disabled.
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 RIP is enabled or disabled.

Showing IP Routing Information

Use the **show ip routes** command to display IP routing information. The syntax and options for this command are as follows:

- **show ip routes** - Displays the entire routing table, including host IP address, next hop, interface, route type, protocol, age (in seconds), and metric.
- **show ip routes firewall** - Displays all firewall routes.
- **show ip routes local** - Displays all local routes.
- **show ip routes ospf** - Displays all OSPF routes.
- **show ip routes rip** - Displays all RIP routes.
- **show ip routes static** - Displays all static routes.
- **show ip routes summary** - Displays the total number of OSPF routes (including a breakdown of Intra, Inter, and Ext routes), RIP routes, local routes, static routes, and firewall routes.
- **show ip routes** *ip_or_host* {**to** *ip_or_host* | *mask_or_prefix*} - Displays information about a route to a destination, a specific route, or routes in a range.

The variables are as follows:

- *ip_or_host* - The IP address of the host or network prefix. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1). The IP address after the **to** keyword is the final IP address in a range.
- *mask_or_prefix* - Subnet address of the specific network. Enter the subnet address in mask or prefix notation (for example, /24).

To show all IP routes in the CSS, enter:

```
# show ip routes
```

Table 6-10 describes the fields in the **show ip routes** command output.

Table 6-10 Field Descriptions for the show ip routes Command

Field	Description
Prefix/length	The IP address and prefix length for the route.
Next hop	The IP address for the next hop.
If	The Index value that identifies the local interface through which the next hop of this route should be reached.
Type	The type of the route entry. The possible types are as follows: • local - Local interface • remote - Remote destination • mgmt - Management interface
Proto	The protocol for the route.
Age	The maximum age of the route.
Metric	The metric cost of the route.

Showing IP Statistics

Use the **show ip statistics** command to display aggregate TCP statistics for the unit. Table 6-11 describes the fields in the **show ip statistics** output.

Table 6-11 Field Descriptions for the show ip statistics Command

Field	Description
UDP Statistics	
Input Datagrams	The total number of flow-related UDP datagrams delivered to UDP users.
No Port Errors	The total number of received UDP datagrams for which there was no application at the destination port.
Output Datagrams	The total number of flow-related UDP datagrams sent from the CSS.
Input Errors	The number of received UDP datagrams that could not be delivered for reasons other than the lack of an application at the destination port.
TCP Statistics	
Retransmit Algorithm	The algorithm used to determine the timeout value for retransmitting unacknowledged octets.
Max Retransmit Time	The maximum value permitted by a TCP implementation for the retransmission timeout, measured in milliseconds.
Active Opens	The number of times TCP connections have made a direct transition to the SYN-SENT state from the Closed state.
Failed Attempts	The number of times TCP connections have made a direct transition to the Closed state from either the SYN-SENT state or the SYN-RCVD state, plus the number of times TCP connections have made a direct transition to the Listen state from the SYN-RCVD state.
Established Conns	The number of TCP connections for which the current state is either Established or Close-Wait.

Table 6-11 Field Descriptions for the show ip statistics Command (continued)

Field	Description
Output Segments	The total number of segments sent, including those on current connections but excluding those containing only retransmitted octets.
Input Errors	The total number of segments received in error (for example, bad TCP checksums).
Min Retransmit Time	The minimum value permitted by a TCP implementation for the retransmission timeout, measured in milliseconds.
Max TCP Connections	The total number of TCP connections that the CSS supports.
Passive Opens	The number of times TCP connections have made a direct transition to the SYN-RCVD state from the LISTEN state.
Resets	The number of times TCP connections have made a direct transition to the CLOSED state from either the ESTABLISHED state or the CLOSE-WAIT state.
Input Segments	The total number of segments received, including those received in error. This count includes segments received on currently established connections.
Retransmit Segments	The total number of segments retransmitted; that is, the number of TCP segments transmitted containing one or more previously transmitted octets.
Output Resets	The number of TCP segments sent containing the RST flag.
ICMP Statistics	
Echo Requests In	The number of received ICMP Echo request messages. Typically, when the CSS receives the ICMP request, both the Echo Requests In and the Echo Replies Out counters increment as a pair for the ICMP request in and ICMP reply out packets.

Table 6-11 Field Descriptions for the show ip statistics Command (continued)

Field	Description
Echo Replies In	The number of received ICMP Echo reply messages. Typically, when the CSS receives an ICMP reply, both the Echo Requests Out and the Echo Replies In counters increment as a pair for the ICMP reply in and ICMP request out packets.
Unreachable	The number of received ICMP Destination Unreachable messages.
Redirect	The number of received ICMP Redirect messages.
Router Solicit	The number of received ICMP router solicitation packets.
Param Problem	The number of received ICMP Parameter Problem messages.
Timestamp Reply	The number of sent ICMP Timestamp Reply messages.
Information Reply	The number of received ICMP information reply packets.
Mask Reply	The number of received ICMP Address Mask Reply messages.
Echo Requests Out	The number of transmitted ICMP Echo request messages. Typically, when the CSS transmits an ICMP request, both the Echo Requests Out and the Echo Replies In counters increment as a pair for the ICMP request out and ICMP reply in packets.
Echo Replies Out	The number of transmitted ICMP Echo reply messages.Typically, when the CSS transmits an ICMP reply, both the Echo Requests In and the Echo Replies Out counters increment as a pair for the ICMP reply out and ICMP request in packets.
Source Quench	The number of received ICMP Source Quench messages.
Router Adv	The number of received ICMP router advertisement packets.

Table 6-11 Field Descriptions for the show ip statistics Command (continued)

Field	Description
Time Exceeded	The number of received ICMP Time Exceeded messages.
Timestamp	The number of sent ICMP Timestamp (request) messages.
Information Request	The number of received ICMP information request packets.
Mask Request	The number of sent ICMP Address Mask Request messages.
Invalid	The number of received bad ICMP type packets.
ARP Statistics	
Requests In	The number of received ARP request packets.
Requests Out	The number of sending ARP request packets.
Duplicate Addr	The number of received ARP packets with a detected duplicate IP address. The duplicate IP address can be the local IP address, VIP, or virtual interface.
Invalid	The number of invalid or bad ARP packets.
Replies In	The number of received ARP reply packets.
Replies Out	The sending ARP reply packet count.
In Off Subnet	The number of received ARP packets with sender or target addresses outside of the subnet range of the receiving interface.
Unresolved	The number of processed IP frames with unresolved next hop MAC addresses.

Showing a Summary of IP Global Statistics

Use the **show ip summary** command to display a summary of IP global statistics. The statistics include data on reachable and total routes, reachable and total hosts, memory in use for each, and total IP routing memory in use.

Table 6-12 describes the fields in the **show ip summary** command output.

Table 6-12 Field Descriptions for the show ip summary Command

Field	Description
Reachable Routes	The current number of reachable routes.
Total Routes	The current number of routes maintained, both reachable and unreachable.
Reachable Hosts	The current number of reachable host entries.
Total Hosts	The current number of host entries, both reachable and unreachable.
Total Memory in use - IP Routing Memory Pool	The total amount of memory in bytes allocated for the IP routing table. When there are no additional free entries in the memory pool, more memory is allocated to the pool.

Configuring the Cisco Discovery Protocol

The Cisco Discovery Protocol (CDP) is a medium-independent protocol that runs over Layer 2 (the data link layer) on the CSS and other Cisco manufactured equipment, such as routers, switches, bridges, and access servers. Use the **cdp** global configuration command to allow the CSS to advertise itself to all other neighboring Cisco CDP-compatible devices on a network. The CSS transmits CDP advertisements to other CDP-compatible devices on the network; the CSS does not listen for CDP messages from the other CDP-compatible devices.

Any Cisco device with CDP support can learn about the CSS by listening to the periodic messages transmitted by the CSS and determining when the CSS is active. Network operators and analysts can use this information for configuration monitoring, topology discovery, and fault diagnosis.

CDP messages contain specific information about the CSS, such as:

- Device ID (CSS base MAC address)
- IP address (CSS management port IP address)
- Ethernet port ID name
- CSS functional capability flag (Router, Transparent Bridge, or Switch)
- CSS software version
- CSS platform

CDP advertisements also include hold time information, which defines the length of time the receiving device is to hold CDP information before discarding it.

This section includes the following topics:

- Enabling CDP
- Setting the CDP Hold Time
- Setting the CDP Transmission Rate
- Showing CDP Information

Enabling CDP

Use the **cdp run** global configuration command to enable CDP transmissions from the CSS to other neighboring Cisco CDP-compatible devices on the network. By default, CDP is disabled for the CSS.

For example:

```
(config)# cdp run
```

To disable CDP transmissions on the CSS, enter:

```
(config)# no cdp run
```

Setting the CDP Hold Time

Use the **cdp holdTime** global configuration command to specify the amount of time a receiving device retains the CDP information sent by the CSS (time-to-live information) before discarding this information. If a neighboring device does not receive a CDP message before the hold time expires, the neighboring device drops the CSS as a neighbor. Valid entries are 10 to 255 seconds. The default is 180 seconds.

To specify a CDP hold time of 255 seconds for the receiving device, enter:

```
(config)# cdp holdTime 255
```

To reset the CDP hold time back to the default value of 180 seconds, enter:

```
(config)# no cdp holdTime
```

Setting the CDP Transmission Rate

Use the **cdp timer** global configuration command to specify the frequency at which the CSS transmits CDP packets to all receiving CDP-compatible devices. Valid entries are 5 to 254 seconds. The default is 60 seconds.

To change the CDP transmission rate for the CSS to 120 seconds, enter:

```
(config)# cdp timer 120
```

To reset the CDP timer to the default rate of 60 seconds, enter:

```
(config)# no cdp timer
```

Showing CDP Information

Use the **show cdp** command to display and verify CDP information for the CSS, such as frequency of transmissions and the hold time for transmitted CSS CDP information.

For example:

```
(config)# show cdp

Global CDP information:
    Sending CDP packets every  60 seconds
    Sending a holdtime value of 16 seconds
    TimeLastCdpSent: 0 days 00:00:30
```

The following example illustrates the CDP output on a Cisco Catalyst 8540 router using the Cisco IOS **show cdp neighbors** command.

```
24-8540-1>show cdp neighbors

Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge S -
Switch, H - Host, I - IGMP, r - Repeater

Device ID          Local Intrfce  Holdtme Capability Platform  Port ID
00-10-58-01-4d-e3   Eth 0            178   R T S     CSS 11501 Eth-Mgmt
SCA043801A5         Eth 0            144   T S       WS-C6009  3/1
25-8540-1           Fas 0/0/7        142   R T       C8540CSR  Fas 0/0/4
25-8540-1           Eth 0            142   R T       C8540CSR  Eth 0
SCA043801HU(bxb11   Eth 0            151   T S       WS-C6009  2/48
00-07-85-43-14-1d   Eth 0            170   R T S     CSS11503  Eth-Mgmt
```

Configuring Spanning-Tree Bridging for the CSS

The CSS supports configuration of Spanning-Tree Protocol (STP) bridging. Spanning-tree bridging detects, and then prevents, loops in the network. Use the **bridge** command to configure global spanning-tree bridging options for the CSS, such as bridge aging time, forward delay time, hello time interval, and maximum age. Make sure you configure the spanning-tree bridging parameters the same on all switches running STP in the network.

Note When connecting a Cisco Catalyst switch to a CSS using an 802.1Q trunk and the STP, the Catalyst runs a spanning-tree instance for each VLAN. When you configure an 802.1Q trunk on an Ethernet interface for the Catalyst switch, the bridge protocol data units (BPDUs) are tagged with the corresponding VLAN ID and the destination MAC address changes from the standard 01-80-C2-00-00-00 to the proprietary 01-00-0c-cc-cc-cd. This modification allows Cisco switches operating in a non-Cisco (a mix of other vendors) 802.1Q trunk environment to maintain spanning-tree states for all VLANs. Although the CSS maintains a spanning-tree instance for each VLAN as well, the CSS uses the standard 01-80-C2-00-00-00 destination MAC address for all BPDUs (tagged or untagged). When you connect a Cisco Catalyst switch to a CSS over an 802.1Q trunk, the result is that neither switch recognizes the other's BPDUs, and both assume root status. If a spanning-tree loop is detected, the Catalyst switch goes into blocking mode on one of its looped ports.

This section includes the following topics:

- Configuring Spanning-Tree Bridge Aging-Time
- Configuring Spanning-Tree Bridge Forward-Time
- Configuring Spanning-Tree Bridge Hello-Time
- Configuring Spanning-Tree Bridge Max-Age
- Configuring Spanning-Tree Bridge Priority
- Disabling Bridge Spanning-Tree
- Showing Bridge Configurations

For details about configuring spanning-tree bridging paremeter for an Ethernet interface or for a trunked Ethernet interface and VLAN pair, refer to Chapter 5, Configuring Interfaces and Circuits.

Configuring Spanning-Tree Bridge Aging-Time

Use the **bridge aging-time** command to set the bridge filtering database aging time for the CSS. The aging time is the timeout period, in seconds, for aging out dynamically learned forwarding information. Enter an integer from 10 to 1000000. The default is 300.

To set the bridge aging time to 600, enter:

```
(config)# bridge aging-time 600
```

To restore the default aging time of 300, enter:

```
(config)# no bridge aging-time
```

Configuring Spanning-Tree Bridge Forward-Time

Use the **bridge forward-time** command to set the bridge forward delay time. The forward time is the delay time, in seconds, that all bridges use for forward delay when this bridge is acting as the root. Enter an integer from 4 to 30. The default is 4.

To set the bridge forward time to 9, enter:

```
(config)# bridge forward-time 9
```

To restore the default delay time of 4, enter:

```
(config)# no bridge forward-time
```

Configuring Spanning-Tree Bridge Hello-Time

Use the **bridge hello-time** command to set the bridge hello time interval. The hello time is the time, in seconds, that all bridges wait before sending a hello packet (when the bridge acts as the root). Enter an integer from 1 to 10. The default is 1.

To set the bridge hello time to 9, enter:

```
(config)# bridge hello-time 9
```

To restore the default hello time interval of 1, enter:

```
(config)# no bridge hello-time
```

Configuring Spanning-Tree Bridge Max-Age

Use the **bridge max-age** command to set the bridge spanning-tree maximum age. The maximum age is the time, in seconds, that protocol information received on a port is stored by the CSS (when a bridge acts as the root). Enter an integer from 6 to 40. The default is 6.

Note Ensure the bridge maximum age is greater than or equal to 2 times (bridge hello-time + 1 second) and less than or equal to 2 times (bridge forward-time - 1 second).

To set the bridge maximum age to 21, enter:

```
(config)# bridge max-age 21
```

To restore the default maximum age of 6, enter:

```
(config)# no bridge max-age
```

Configuring Spanning-Tree Bridge Priority

To set the priority that spanning tree uses to choose the root bridge in the network, use the global **bridge priority** command. In spanning tree, the 2-octet field is prepended to the 6-octet MAC address to form an 8-octet bridge identifier. The device with the lowest bridge identifier is considered the highest priority bridge and becomes the root bridge. The range for bridge priority is 0 to 65535. The default is 32768.

For example:

```
(config)# bridge priority 1700
```

To restore the bridge priority to the default of 32768, enter:

```
(config)# no bridge priority
```

Disabling Bridge Spanning-Tree

Spanning-tree bridging is enabled by default. When you disable spanning-tree bridging, the CSS forwards all multicast traffic, including bridge protocol data units (BPDUs) for the bridge multicast group and for trunked VLANs. The CSS can still operate in an 802.1Q spanning-tree environment as long as you do not require that the CSS put any of its ports into a blocking state.

To disable spanning-tree bridging, enter:

```
(config)# bridge spanning-tree disable
```

Caution Disabling spanning-tree bridging may make your network susceptible to packet storms.

To reenable spanning-tree bridging, enter:

```
(config)# bridge spanning-tree enable
```

Showing Bridge Configurations

Use the **show bridge forwarding** command to display bridge forwarding information. Table 6-13 describes the fields in the **show bridge forwarding** command output.

Table 6-13 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 used for bridge forwarding

Use the **show bridge status** command to display bridge status information. Table 6-14 describes the fields in the **show bridge status** output.

Table 6-14 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, during which the host times out root information.
Root Hello Time	The interval, in seconds, during which the root bridge broadcasts its hello message to other devices.
Root Fwd Delay	The delay time, in seconds, that the root bridge uses for forward delay.
Designated Root	The bridge ID for the designated root.
Bridge ID	The bridge ID of the bridge.
Port	The port ID.
State	The state of the port. The possible states are as follows: • Block - The blocking state. A port enters the blocking state after CSS 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 the Spanning-Tree Protocol. A port in the disabled state is non operational.

Table 6-14 Field Descriptions for the show bridge status Command (continued)

Field	Description
Designated Bridge	The bridge ID for the designated bridge.
Designated Root	The bridge ID for the designated root.
Root Cost	The cost of the root.
Port Cost	The cost of the port.
Desg Port	Designated port.

Configuring the DHCP Relay Agent

The Dynamic Host Configuration Protocol (DHCP) servers provide configuration parameters to DHCP clients. When DHCP clients and associated servers do not reside on the same IP network or subnet, a DHCP relay agent can transfer DHCP messages between them. To configure a DHCP relay agent on a CSS, define DHCP server destinations on a circuit and enable the DHCP relay agent on the circuit.

You must first assign an IP address on the circuit to be able to configure the DHCP relay agent for the circuit. Use the **ip address** command in the specific circuit mode to assign the IP address and a subnet mask. For example:

```
(config-circuit[VLAN2])# ip address 178.3.6.53/8
```

This section includes the following topics:

- Adding a DHCP Destination on a Circuit
- Enabling and Disabling DHCP on the Circuit
- Defining the Hops Field Value for Forwarding DHCP Messages
- Displaying the DHCP Relay Configuration

Adding a DHCP Destination on a Circuit

A CSS circuit acts as the DHCP relay agent. For each circuit on the CSS, you can configure a maximum of five DHCP destinations. The initial DHCP broadcast request is sent to all of the configured destinations.

Do not configure a relay destination on a circuit when the relay destination is directly connected to or reachable from one of the ports on the same circuit. In this case, the DHCP packets reach the relay destination through normal broadcast and a relay agent is not required.

Use the **dhcp relay-to** command to specify the DHCP relay destination address. This command is available in circuit configuration mode. Enter an IP address in dotted-decimal notation.

For example, to add a destination address of 192.168.22.25 to a DHCP server, enter:

```
(config-circuit[VLAN2])# dhcp relay-to 192.168.22.25
```

To remove the relay destination address, enter:

```
(config-circuit[VLAN2])# no dhcp relay-to 192.168.22.25
```

Enabling and Disabling DHCP on the Circuit

After you enable the DHCP relay agent on the CSS circuit, the CSS transfers DHCP messages between DHCP clients and servers. Use the **dhcp-relay-agent** command to enable the agent on the circuit. This command is available in circuit configuration mode.

For example:

```
(config-circuit[VLAN2])# dhcp-relay-agent
```

To disable the DHCP relay agent on the circuit, enter:

```
(config-circuit[VLAN2])# no dhcp-relay-agent
```

Defining the Hops Field Value for Forwarding DHCP Messages

The CSS forwards or discards a DHCP message based on the hops field value in the BOOTP header. When messages have values in the hops fields that exceed the maximum value set on the CSS, the CSS discards the message. Use the **dhcp-agent max-hops** global configuration command to set the maximum allowable number in the hops field. By default, the maximum allowable number is 4. You can set a number from 1 to 15.

For example, to set the maximum allowable value of 10, enter:

```
(config)# dhcp-agent max-hops 10
```

To reset the maximum allowable number in the hops field to the default of 4, enter:

```
(config)# no dhcp-agent max-hops
```

Displaying the DHCP Relay Configuration

Use the **show dhcp-relay-agent global** command to display the DHCP configuration information on a CSS. This command is available in all modes. For example:

```
# show dhcp-relay-agent global
```

Table 6-15 describes the fields in the **show dhcp-relay-agent global** command output.

Table 6-15 Field Descriptions for the show dhcp-relay-agent global Command

Field	Description
Max Hops	The maximum allowable number in the hops field of the BOOTP header. The CSS does not forward packets with headers that contain a larger number.
Number of circuits configured for DHCP	The number of CSS circuits configured for DHCP.
Circuit	The circuit configured for DHCP.
IfAddress	The interface address for the circuit.

Table 6-15 Field Descriptions for the show dhcp-relay-agent global Command

Field	Description
DHCP State	The DHCP relay agent state on the circuit (Enabled or Disabled).
Relay destination	The DHCP relay destination address for the server. Each circuit can have five destination addresses.

Where to Go Next

Chapter 7, Configuring Open Shortest Path First (OSPF) describes how to configure and view information for the OSPF protocol.

Configuring Open Shortest Path First (OSPF)

This chapter provides configuration and viewing information for the Open Shortest Path First (OSPF) protocol. Information in this chapter applies to all CSS models, except where noted.

Note The CSS supports OSPF Version 2, as defined in RFC 2178. For detailed information about OSPF MIB objects, refer to RFC 1850.

This chapter contains the following major sections:

- OSPF Overview
- CSS OSPF Quick Configuration
- Configuring OSPF on the CSS
- Configuring OSPF on a CSS IP Interface
- Showing OSPF Information
- OSPF Configuration in a Startup-Config File

OSPF Overview

OSPF is a link-state routing protocol that:

- Provides network topology discovery within a group of routers and networks called an autonomous system (AS)
- Calculates the shortest path to destinations within the AS

As a link-state protocol, OSPF routers flood any change in routing information throughout the network. This action differs from a distance vector protocol, such as RIP, which periodically exchanges routing information only with neighboring devices.

Within an AS, each OSPF router builds and synchronizes a database of the AS network topology. The routers synchronize their databases by requesting information from other AS routers. Each router sends its information as link-state advertisements (LSAs) that include information about the state of each router and link in the AS. A link is an interface on the router. The state of the link is the description of the interface, including the router's IP address and subnet mask, and its relationship to the neighboring router.

Then, the router uses its database and the Shortest Path First (SPF) algorithm to calculate the shortest path to every destination in the AS and stores this information in a dynamic table. When changes occur, the router calculates new paths.

The CSS, operating as an OSPF router, provides:

- Intra-area route support for routing in a single area between other OSPF routers
- Inter-area route support for routing between multiple OSPF areas
- Route summarization between areas as an Area Border Router (ABR)
- Stub area and AS boundary router support
- Redistribution of local, RIP, static, and firewall routes into an OSPF domain
- Advertisement of VIP addresses for content as AS external routes
- Simple authentication

TThis section includes the following topics:

- OSPF Routing Hierarchy
- Link-State Databases

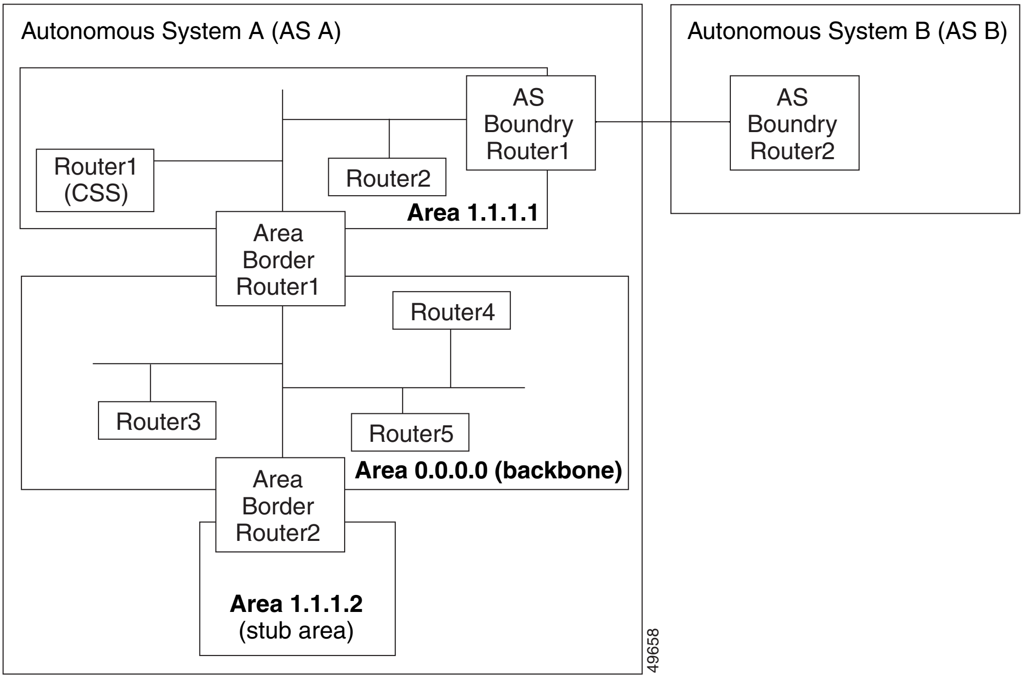
OSPF Routing Hierarchy

The OSPF routing hierarchy includes the following functions:

- Autonomous systems
- Areas, including the backbone and stub areas
- Area Border Routers (ABRs)
- Autonomous System Boundary Routers (ASBRs)

Figure 7-1 illustrates an OSPF network topology.

Figure 7-1 Basic OSPF Network Topology



Autonomous System

The autonomous system (AS) is a collection of networks, under the same administrative control, that share the same routing information with each other. An AS is also referred to as a routing domain. Figure 7-1 shows two ASs: AS A and AS B. An AS can consist of one or more OSPF areas.

Areas

Areas allow the subdivision of an AS into smaller, more manageable networks or sets of adjacent networks. As shown in Figure 7-1, AS A consists of three areas: area 0.0.0.0, area 1.1.1.1, and area 1.1.1.2.

OSPF hides the topology of an area from the rest of the AS. An area's network topology is visible only to routers inside that area; the network topology is not visible to routers outside the area. When OSPF routing is within an area, this is called intra-area routing. This routing limits the amount of link-state information flooding onto the network, thereby reducing routing traffic. OSPF routing also reduces the size of the topology information in each router, which conserves processing and memory requirements in each router.

Conversely, the routers within an area cannot see detailed network structures outside the area. Because of this restriction of topological information, you can control traffic flow between areas and reduce routing traffic when the entire autonomous system is a single routing domain.

Backbone Area

A backbone area is responsible for distributing routing information between the areas of an autonomous system. When OSPF routing occurs outside of an area, this is called inter-area routing.

The backbone itself has all the properties of an area. It consists of ABRs, and routers and networks only on the backbone. As shown in Figure 7-1, area 0.0.0.0 is an OSPF backbone area. Note that a designated OSPF backbone area has a reserved ID of 0.0.0.0.

Area Border Routers

ABRs have multiple interfaces that connect directly to networks in two or more areas. An ABR runs a separate copy of the OSPF algorithm and maintains separate routing data for each area that is connected to it, including the backbone area. ABRs also send configuration summaries for their attached areas to the backbone area, which distributes this information to other OSPF areas in the autonomous system. In Figure 7-1, there are two ABRs. ABR 1 interfaces area 1.1.1.1 to the backbone area. ABR 2 interfaces the backbone area to area 1.1.1.2, a stub area.

Note ABRs are always backbone routers. You must configure ABRs to the backbone area.

Stub Area

A stub area is an area that does not accept or distribute detailed network information external to the area. A stub area has only one router that interfaces the area to the rest of the AS. The ABR attached to the stub area advertises a single default external route into the area. Routers within a stub area use this route for destinations outside the autonomous system, as well as for inter-area routes. This relationship conserves LSA database space that would otherwise be used to store external LSAs flooded into the area. As shown in Figure 7-1, area 1.1.1.2 is a stub area that is reached only through ABR 2.

Autonomous System Boundary Routers

ASBRs provide connectivity from one autonomous system to another system. ASBRs exchange their autonomous system routing information with boundary routers in other autonomous systems. Every router inside an autonomous system knows how to reach the boundary routers for its autonomous system.

ASBRs can import external routing information from other protocols like RIP and redistribute them as AS-external LSAs to the OSPF network. If the CSS is an ASBR, you can configure it to advertise VIP addresses for content as AS external routes. In this way, ASBRs flood information about external networks to routers within the OSPF network.

ASBR routes can be advertised as type1 or type2 ASE. The difference between type1 and type2 is how the cost is calculated. For a type2 ASE, only the external cost (metric) is used when comparing multiple paths to the same destination. For type1 ASE, the combination of the external cost and the cost to reach the ASBR is used.

Link-State Databases

OSPF routers advertise routes using LSAs. The link-state database stores the LSAs from routers throughout the area. The advertisements depict the topology of the autonomous system. They could include:

- Router links that describe the state and cost of each router's interface to an area
- Network links from the designated router (see the "Setting the Priority of the CSS" section) that describe all routes on a segment for multi-access segments with more than one attached router
- Summarized links from ABRs that describe networks in the AS but outside an area
- External links from ASBRs that describe destinations external to the AS

All routers that are connected to an area maintain identical routing databases about the area. Routers that are connected to multiple areas maintain a separate routing database for each attached area.

Instead of each router sending routing information to every other router on the network, OSPF routers establish adjacencies among neighboring routers. When the link-state databases of two neighboring routers are synchronized, they are considered adjacent.

OSPF routers collect raw topological data from the LSAs that they receive. Each router then prunes this data down to a tree of the shortest network paths centered on itself. The router examines the total cost to reach each router or network node in its domain. By discarding all but the lowest-cost path to each destination, the router builds a shortest-path tree to each destination, which it uses until the network topology changes. It is possible to have multiple lowest-cost paths to a destination.

CSS OSPF Quick Configuration

This section includes the following topics:

- Global OSPF Quick Configuration
- OSPF IP Interface Quick Configuration
- Verifying Your Quick Configuration

Global OSPF Quick Configuration

To perform the global OSPF configuration for the CSS, see the steps in Table 7-1. In the most basic global configuration, where the CSS functions as a router in the OSPF backbone area, you need to perform only steps 1 and 2 to:

- Define the CSS router ID
- Enable OSPF

Optionally, you can define the CSS:

- In an area other than the backbone, including a stub area.
- As an ABR, by configuring route summarization.
- As an ASBR, to advertise non-OSPF routes through OSPF, as AS-external routes such as static and RIP routes. You could also advertise VIP addresses for content as AS external routes.

After performing the global OSPF configuration, you must configure an OSPF IP interface (see the "OSPF IP Interface Quick Configuration" section) before the CSS can participate in OSPF routing. For more information on configuring global OSPF parameters, see the "Configuring OSPF on the CSS" section.

Table 7-1 CSS Global OSPF Quick Configuration

Task and Command Example
1. Configure the area router ID for the CSS in global configuration mode. In this example, the CSS router ID is 121.23.21.1. `(config) ospf router-id 121.23.21.1`
2. Enable OSPF on the CSS. `(config) ospf enable`
3. (Optional) If the CSS area is other than the backbone area, enter the area ID for the CSS. In this example, the area ID is 1.1.1.1. `(config) ospf area 1.1.1.1` The default ID is 0.0.0.0 for the backbone area. To define a stub area, enter the stub option after the area ID.
4. (Optional) If you want the CSS to advertise external routes, define the CSS as an AS boundary router. For example: `(config) ospf as-boundary`
5. (Optional) If the CSS is an ABR, you can advertise VIP addresses for content as OSPF ASE routes. To advertise the VIP address 192.168.4.15 with a default cost of 1 and the default type of ASE type2, enter: `(config) ospf advertise 192.168.4.15 255.255.255.255`
6. (Optional) To advertise routes other than OSPF, such as a firewall, local, RIP or static route, configure OSPF to redistribute routes from the specific protocol. To advertise static routes through OSPF with a default cost of 1 and default type of ASE type2, enter: `(config) ospf redistribute static`

After you complete the global OSPF configuration, configure OSPF on CSS IP interfaces as described in the "OSPF IP Interface Quick Configuration" section.

OSPF IP Interface Quick Configuration

To configure OSPF on a CSS IP interface, see the steps in Table 7-2. In the most basic IP interface configuration, you need to perform only steps 1 through 4, and step 7 to:

- Assign OSPF to the IP interface
- Associate OSPF with the globally defined area, if this is an area other than the backbone area (0.0.0.0)
- Enable OSPF on the interface

This configuration example assumes you will accept the default OSPF configuration settings for the interface, except the router priority. The interface OSPF configuration settings include:

- Intervals for the hello packet, LSA retransmission, and link-state update packet
- Authentication password
- CSS router priority
- Interface cost

For more information on configuring these OSPF IP interface settings, see the "Configuring OSPF on a CSS IP Interface"section.

Table 7-2 Quick Configuration for OSPF on a CSS Interface

Task and Command Example
1. Access global configuration mode. Enter: `# config`

2. Access the circuit configuration mode for a preconfigured circuit on which you want to create the IP interface. For example, if circuit VLAN6 already exists, enter:
`(config)# circuit VLAN6`
`(config-circuit[VLAN6])#`
Note Refer to Chapter 5, Configuring Interfaces and Circuits for information on how to configure the CSS interfaces and circuits and the bridge interfaces to VLANs.

Table 7-2 Quick Configuration for OSPF on a CSS Interface (continued)

Task and Command Example

3. Create the IP interface to the circuit. To create an IP address of 3.1.2.2 with a subnet mask of /24, enter:
`(config-circuit[VLAN6])# `**`ip address 3.1.2.2/24`**
`Create ip interface <3.1.2.2>, [y/n]: `**`y`**

4. Configure the IP interface as an OSPF interface. Enter: `(config-circuit-ip[VLAN6-3.1.2.2])# `**`ospf`**
5. (Optional) If the globally configured area is other than the backbone area, enter the configured area ID. In this example, the globally configured area ID is 1.1.1.1. `(config-circuit-ip[VLAN6-3.1.2.2]) `**`ospf area 1.1.1.1`**
6. (Optional) With a default setting of 1, the CSS is set to a priority that allows it to become the designated router. If you do not want the CSS to become the designated router, you can change its priority or disable it from eligibility. For example, if you want the CSS to be ineligible to become a designated router, enter: `(config-circuit-ip[VLAN6-3.1.2.2])# `**`ospf priority 0`** For more information on designated routers, see the "Setting the Priority of the CSS" section.
7. Enable OSPF on the interface. Enter: `(config-circuit-ip[VLAN6-3.1.2.2])# `**`ospf enable`**

Verifying Your Quick Configuration

To verify the OSPF global and interface quick configurations, use the **show ospf** command and its options. For example:

- To show the OSPF global configuration, use the **show ospf global** command. For example:

  ```
  # show ospf global
  ```

  If the Admin Status field is disabled, use the **ospf enable** command to enable OSPF.

- To show the route redistribution policy into OSPF, use the **show ospf redistribute** command. To show the configured static route redistribution policy, enter:

  ```
  # show ospf redistribute
  ```

- To show the VIP addresses advertised as ASE routes, use the **show ospf advertise** command. For example:

  ```
  # show ospf advertise
  ```

- To view the CSS IP interface configuration, use the **show ospf interfaces** command. For example:

  ```
  # show ospf interfaces
  ```

Configuring OSPF on the CSS

This section includes the following topics:

- Configuring the OSPF Router ID
- Enabling OSPF
- Configuring an Area
- Configuring Equal-Cost Routes
- Configuring Summarized Routes at an ABR
- Configuring the CSS as an Autonomous System Boundary Router

Configuring the OSPF Router ID

Before you enable OSPF on the CSS, configure the router ID. Assigning a router ID to the CSS uniquely identifies it to other routers within the autonomous system. In addition, in the case of a priority tie when determining which router is the designated router, the ID serves as a tie-breaker in the designated router election. For more information on designated routers, see the "Setting the Priority of the CSS" section.

Use the **ospf router-id** command to configure the OSPF router ID for the CSS. A router ID is a 32-bit number in dotted-decimal notation.

To assign the router ID of 121.23.21.1 to the CSS, enter:

```
(config)# ospf router-id 121.23.21.1
```

Note If OSPF is globally enabled, use the **no** form of the **ospf enable** command to disable OSPF and change the router ID.

To delete the router ID on the CSS, disable OSPF and enter:

```
(config)# no ospf router-id
```

Enabling OSPF

After you assign the router ID to the CSS, globally enable OSPF on the CSS. Use the **ospf enable** command to enable OSPF. For example:

```
(config)# ospf enable
```

To disable OSPF, enter:

```
(config)# no ospf enable
```

Configuring an Area

By default, the CSS is configured to the backbone area automatically. The backbone area has a reserved ID of 0.0.0.0. If the CSS is part of an area other than the backbone area, assign the CSS to that area.

Use the **ospf area** command to assign an area. Enter the ID in dotted-decimal notation (for example, 0.0.0.1). Although an area ID has the same form as an IP address, the area ID address space is its own distinct address space.

For example, if the CSS is in area 0.0.0.1, enter:

```
(config)# ospf area 0.0.0.1
```

If the CSS is in a stub area, include the **stub** option.

For example, if area 0.0.0.1 is a stub area, enter:

```
(config)# ospf area 0.0.0.1 stub
```

Optionally, for a stub area you can also:

- Set a metric for the default route advertised in the stub area.
- Propagate summary LSAs into the stub area.

To set a metric for the default route advertised in the stub area, include the **default-metric** option. By default, the metric equals the smallest metric among the interfaces to other areas. You can assign an integer from 1 to 16777215.

For example, to assign a metric of 200, enter:

```
(config)# ospf area 0.0.0.1 stub default-metric 200
```

To propagate summary LSAs in the stub area, include the **send-summaries** option. For example:

```
(config)# ospf area 0.0.0.1 stub send-summaries
```

Removing an Area

To remove an OSPF area, disable OSPF, then use the **no** form of the **ospf area** command. For example:

```
(config)# no ospf enable
(config)# no ospf area 0.0.0.1
```

Configuring Equal-Cost Routes

By default, the OSPF CSS is configured to use 15 equal-cost routes. Use the **ospf equal-cost** command to change the number of routes. Enter a number from 1 to 15.

To configure 10 equal-cost routes for use by the CSS, enter:

```
(config)# ospf equal-cost 10
```

To reset the equal-cost routes to its default value of 15, enter:

```
(config)# no ospf equal-cost
```

Configuring Summarized Routes at an ABR

If the CSS is an ABR, you can configure it to advertise a single summary route or network ranges that cover all the individual networks within the specified range. This summarization helps control routing table sizes and prevents the constant changing of routes whenever an interface within an area comes online or goes offline. These route changes do not cause route changes in backbone ABRs and other area routers.

Use the **ospf range** command to specify the IP address range to summarize routes at the ABR. This summarization applies to inter-area paths that are paths to destinations in other OSPF areas. You can also determine whether you want to advertise this range. Disable OSPF before you enter the **ospf range** command.

Define an address range by specifying an IP address and subnet mask that represents networks in the area being summarized. Enter the IP address and subnet mask in dotted-decimal notation (for example, 192.168.128.0 255.255.224.0). You can also enter the mask in CIDR bit-count notation format (for example, /24).

To configure the CSS as an ABR with an area ID of 0.1.0.1 with a collection of destinations between 192.168.0.0 and 192.168.255.255, enter:

```
(config)# no ospf enable
(config)# ospf range 0.1.0.1 192.168.0.0 255.255.0.0
```

To remove the range, enter:

```
(config)# no ospf range 0.1.0.1 192.168.0.0 255.255.0.0
```

By default, the ABR advertises this range. If you want to hide the range from the rest of the AS, include the **block** option. For example:

```
(config)# ospf range 0.1.0.1 192.168.0.0 255.255.0.0 block
```

Configuring the CSS as an Autonomous System Boundary Router

Use the **ospf as-boundary** command if you want the CSS to be an ASBR that exchanges routing information with routers belonging to other autonomous systems. Disable OSPF before you enter the **ospf as-boundary** command.

For example:

```
(config)# no ospf enable
(config)# ospf as-boundary
```

To remove the CSS as an AS boundary router, enter:

```
(config)# no ospf as-boundary
```

To advertise a route as OSPF ASE through all OSPF interfaces or generate a default route, see the following sections.

- Advertising a Route as an OSPF ASE Route
- Advertising a Default ASE Route
- Advertising Other Routes Through OSPF

Advertising a Route as an OSPF ASE Route

The ASBR can perform external route summarization to consolidate multiple routes into a single advertisement. For a CSS, this consolidation is useful when you want to advertise VIP addresses for content as OSPF AS external (ASE) through all OSPF interfaces. Use the **ospf advertise** command to advertise a route as OSPF ASE through all OSPF interfaces. To stop the advertisement of the route, use the **no** form of the **ospf advertise** command (as described later in this section).

When you configure OSPF advertising of an IP address as an ASE route, note that:

- If the advertised IP address is the redundant VIP in a VIP redundancy virtual router configuration environment, OSPF advertises the VIP address when the virtual router state is Master, but OSPF stops advertising this VIP address when the virtual router state is Down or Backup.
- If the advertised IP address is a service IP in a service record configuration, OSPF advertises the IP address when the service state is Alive, Dying, or Suspend, but OSPF stops advertising this IP address when the service state is Down.
- If the advertised IP address is the service IP for a critical service, in which the virtual router has a dependency on the critical service, OSPF treats the IP address the same as the service IP described above. The VIP redundancy state does not have an impact on whether OSPF advertises the IP address; only the service state determines the impact on OSPF advertising.
- If OSPF is configured to advertise an IP address that does not match any service IP and redundant VIP, OSPF advertises the IP address all the time. If this address is later used as a service IP, OSPF determines whether to advertise the IP address based on the service state. When the service state for the service IP is Down, OSPF does not advertise the service IP. In addition, OSPF does not revert the service IP back to the original IP address and resume advertising it when the IP address does not overlap with a service IP.

First, before you enter the **ospf advertise** command, configure the CSS as an ASBR. For more information, see the "Configuring the CSS as an Autonomous System Boundary Router" section.

Define an address range for the **ospf advertise** command by specifying an IP address and subnet mask that represents networks in the area being summarized. Enter the IP address and subnet mask in dotted-decimal notation (for example, 192.168.128.0 255.255.224.0). You can also enter the mask in CIDR bit-count notation format (for example, /24).

For example, to advertise VIP addresses from 192.168.44.0 to 192.168.44.255, define the range by entering the IP address and subnet mask of 192.168.44.0 255.255.255.0:

```
(config)# ospf advertise 192.168.44.0 255.255.255.0
```

Optionally, you can define any of the following:

- The network cost for the route by including the **metric** option. Enter a number from 1 to 16777215. The default is 1.
- A 32-bit tag value to advertise each external route by including the **tag** option. The 32-bit tag value is not used by the OSPF protocol itself. You can use the tag value to communicate information between ASBRs.
- The advertised routes as ASE type1 by including the **type1** option. By default, the type is ASE type2. The difference between type1 and type2 is how the cost is calculated. For a type2 ASE, only the external cost (metric) is used when comparing multiple paths to the same destination. For type1 ASE, the combination of the external cost and the cost to reach the ASBR is used.

For example:

```
(config)# ospf advertise 193.23.44.0 255.255.255.0 metric 3 type1
```

To stop advertising of the route as OSPF ASE through all OSPF interfaces, enter:

```
(config)# no ospf advertise 193.23.44.255.255.255.0
```

The following running configuration example illustrates the **ospf advertise** command for OSPF advertising of VIP addresses and an IP address. Comments are preceded by an exclamation point (!).

```
!*************************** GLOBAL ***************************
ospf enable

ospf advertise 1.1.1.10
!advertise redundant VIP
ospf advertise 2.1.1.1
!advertise IP address of service s1
ospf advertise 1.1.1.100
!advertise IP address of critical service c100
ospf advertise 99.99.99.99
!advertise simple IP address, not tied to anything
record

!************************** CIRCUIT **************************
circuit VLAN1

ip address 1.1.1.200 255.0.0.0
ip virtual-router 1
ip redundant-vip 1 1.1.1.10
!redundant VIP
ip critical-service 1 c100

!************************** SERVICE **************************
service c100
ip address 1.1.1.100
!IP address for critical service
active

service s1
ip address 2.1.1.1
!IP address for service s1
keepalive method get
keepalive type http
active

service s2
ip address 2.1.1.2
keepalive method get
keepalive type http
active
```

```
!************************* OWNER **************************
owner admin1

content r1
add service s1
add service s2
vip address 1.1.1.10
!redundant VIP equals content VIP

active
```

Advertising a Default ASE Route

Routers use default routes when no additional routes exist to a particular AS external destination. By default, an ASBR does not generate a default route into the OSPF routing domain. Use the **ospf default** command to force the CSS to generate a default ASE route and advertise the route through OSPF.

Before you enter the **ospf default** command, configure the CSS as an ASBR. For more information, see the "Configuring the CSS as an Autonomous System Boundary Router" section.

For example:

```
(config)# ospf default
```

Optionally, you can define any of the following:

- The network cost for an OSPF default route by including the **metric** option. If a default route metric is defined, the router advertises itself as the default router to the area. Enter a number from 1 to 16,777,215. The default is 1.
- A 32-bit tag value to advertise each external route by including the **tag** option. The 32-bit tag value is not used by the OSPF protocol itself. You can use the tag value to communicate information between ASBRs.
- The advertised routes as ASE type1 by including the **type1** option. By default, the type is ASE type2. The difference between type1 and type2 is how the cost is calculated. For a type2 ASE, only the external cost (metric) is used when comparing multiple paths to the same destination. For type 1 ASE, the combination of the external cost and the cost to reach the ASBR is used.

For example:

```
(config)# ospf default metric 10 type1
```

To stop advertising the default ASE routes originated through OSPF, enter:

```
(config)# no ospf default
```

Advertising Other Routes Through OSPF

Use the **ospf redistribute** command to advertise routes from other protocols, such as firewall, local, RIP, and static routes through OSPF. Redistribution of these routes makes them OSPF external routes.

To redistribute routes from other protocols, include one of the following options:

- **firewall** - Advertises firewall routes through OSPF
- **local** - Advertises local routes (interfaces *not* running OSPF)
- **rip** - Advertises RIP routes through OSPF
- **static** - Advertises static routes configured for the Ethernet interface ports. The **ospf redistribute static** command does not advertise static routes configured for the Ethernet management port.

To advertise a firewall route, enter:

```
(config)# ospf redistribute firewall
```

Optionally, you can define any of the following:

- The network cost for the route by including the **metric** option. Enter a number from 1 to 16,777,215. The default is 1.
- A 32-bit tag value to advertise each external route by including the **tag** option. The 32-bit tag value is not used by the OSPF protocol itself. You can use the tag value to communicate information between AS boundary routers.
- The advertised routes as ASE type1 by including the **type1** option. By default, the type is ASE type2. The difference between type1 and type2 is how the cost is calculated. For a type2 ASE, only the external cost (metric) is considered when comparing multiple paths to the same destination. For type1 ASE, the combination of the external cost and the cost to reach the ASBR is used.

For example:

```
(config)# ospf redistribute rip metric 3 type1
```

To stop advertising the RIP routes via OSPF, enter:

```
(config)# no ospf redistribute rip
```

Configuring OSPF on a CSS IP Interface

When you configure a CSS IP interface as an OSPF interface, you define its behavior and role within the OSPF routing domain. This section includes the following topics:

- Configuring the CSS IP Interface as an OSPF Interface
- Assigning an OSPF Area to the Interface
- Enabling OSPF on the Interface
- Configuring the Interface Attributes

Configuring the CSS IP Interface as an OSPF Interface

An OSPF interface is an IP interface that you configure to send and receive OSPF traffic. To configure the CSS IP interface as an OSPF interface, use the **ospf** command.

Note You must enter the **ospf** command before the **ospf enable** command can take effect.

To configure the CSS IP interface as an OSPF interface:

1. Access the circuit configuration mode for the preconfigured circuit on which you want to create the IP interface. For example, if circuit VLAN6 already exists, enter:

```
(config)# circuit VLAN6
(config-circuit[VLAN6])#
```

Note Refer to Chapter 5, Configuring Interfaces and Circuits for information on how to configure the CSS interfaces and circuits, and bridge interfaces to VLANs.

2. Create the IP interface to the circuit. To create an IP address of 3.1.2.2, enter:

```
(config-circuit[VLAN6])# ip address 3.1.2.2/24
Create ip interface <3.1.2.2>, [y/n]:y
```

3. Configure this circuit as an OSPF circuit. Enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# ospf
```

Assigning an OSPF Area to the Interface

After you configure the IP interface as an OSPF interface, assign it to the area that you globally configured to the CSS. The default area is the backbone area with the ID of 0.0.0.0. If the area is other than the backbone, use the **ospf area** command to assign the interface to an OSPF area. For example, if the area is 0.0.0.1, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# ospf area 0.0.0.1
```

To reset the interface to the default backbone area, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# no ospf area
```

Enabling OSPF on the Interface

Note If you need to configure the interface attributes as described in the “Configuring the Interface Attributes” section, do not enable OSPF on the IP interface until you finish configuring the attributes.

Use the **ospf enable** command to enable OSPF on the IP interface. By default, OSPF is disabled on an IP interface. For example:

```
(config-circuit-ip[VLAN6-3.1.2.2])# ospf enable
```

To disable OSPF on the interface, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# no ospf enable
```

Configuring the Interface Attributes

The OSPF interface attributes are set to a series of default values. You can elect to use these values for the CSS IP interface or configure your own settings. This section includes the following topics:

- Setting the Cost
- Setting the Dead Router Interval
- Setting the Hello Packet Interval
- Setting the Password
- Setting the Poll Interval
- Setting the Priority of the CSS
- Setting the Retransmission Interval
- Setting the Transit-Link Delay

Setting the Cost

Use the **ospf cost** command to set the cost for sending a data packet on this interface. The cost for the interface is a number from 0 to 65535. The default value of the cost for a given type of circuit is 108/interface speed. For a Gigabit Ethernet interface, the value is 1. For a 10/100-Mbps Fast Ethernet interface, the value is 10.

For example, to set a cost of 25, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# ospf cost 25
```

To reset the packet cost for the interface to the default value, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# no ospf cost
```

Setting the Dead Router Interval

The interface declares a neighbor router is dead if the interface does not receive hello packets from the router before the dead interval expires. Use the **ospf dead** command to set the dead router interval for an interface. The dead router interval is in seconds. This value must be a multiple of the hello interval, and the value must be the same for all routers attached to a common network. Enter a number from 1 to 2,147,483,647. The default is 40.

For example, to set the dead router interval to 100 seconds, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# ospf dead 100
```

To reset the dead router interval to its default of 40 seconds, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# no ospf dead
```

Setting the Hello Packet Interval

Router interfaces periodically transmit hello packets to identify and maintain communications with their neighbors. When a router detects its own address in another router's hello packet, the two routers establish two-way communications as neighbors.

The hello interval is the length of time, in seconds, between hello packets that the interface sends to its neighbor routers. The hello interval must be the same value for all routers attached to a common network. Use the **ospf hello** command to set the hello interval for the IP interface. Enter an integer from 1 to 65535. The default is 10 seconds.

To set a hello interval of 25 seconds, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# ospf hello 25
```

To reset the hello interval to the default value of 10 seconds, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# no ospf hello
```

Setting the Password

All OSPF protocol exchanges can be authenticated to ensure only known, trusted routers participate in routing updates. The OSPF password is used for authentication of all OSPF protocol exchanges.

Use the **ospf password** command to set the password for an interface. This password must be the same for all routers attached to a common network. Enter a quoted text string with a maximum of eight characters.

For example, to set the password of *quota*, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# ospf password “quota”
```

To remove the OSPF password from the interface, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# no ospf password
```

Setting the Poll Interval

The poll interval is the length of time, in seconds, between the transmittal of hello packets by the CSS to an assumed inactive neighbor router in a non-broadcast, multi-access network. Use the **ospf poll** command to set the poll interval for the interface. The poll interval should be a value that is greater than the hello time interval. Enter a number from 1 to 2,147,483,647. The default is 120 seconds.

Note The **ospf poll** command has no effect when you operate the CSS over a broadcast LAN (that is, an Ethernet network).

For example, to set the poll interval to 200 seconds, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# ospf poll 200
```

To reset the poll interval to the default value of 120 seconds, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# no ospf poll
```

Setting the Priority of the CSS

To avoid the need for each router on a LAN to talk to every router on a network that has more than two attached routers, one router is elected as the designated router. Designated routers advertise network link states for attached network segments. An LSA lists all routers that are connected to a segment.

The priority determines which router is the designated router. The router with the highest priority becomes the designated router. In case of a tie, routers use their router ID as a tie breaker.

Use the **ospf priority** command to set the router priority for the interface. The priority of the interface is an integer from 0 to 255. The default is 1, which is the highest router priority. A value of 0 signifies that the CSS is not eligible to become the designated router on a particular network.

If a designated router exists on the network, it remains the designated router regardless of its router priority.

To make the interface ineligible to become a designated router, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# ospf priority 0
```

To reset the router priority to the default value of 1, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# no ospf priority
```

Setting the Retransmission Interval

The retransmission interval is the number of seconds between link-state advertisement retransmissions for adjacencies belonging to an interface. OSPF creates adjacencies between neighboring routers for the purpose of exchanging routing information. The CSS also uses the interval when retransmitting database descriptions and link-state request packets.

Use the **ospf retransmit** command to set the retransmit interval for the interface. Enter a number from 1 to 3600 seconds (1 hour). The default is 5 seconds.

To set the retransmission interval to 10 seconds, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# ospf retransmit 10
```

To reset the retransmit interval to the default value of 5 seconds, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# no ospf retransmit
```

Setting the Transit-Link Delay

Transit delay is the estimated number of seconds the CSS waits to transmit a link-state update packet over the OSPF interface. Use the **ospf transit-delay** command to set the transit delay for an interface. Enter a number from 0 to 3600 seconds (1 hour). The default is 1 second.

To set the transit delay to 3 seconds, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# ospf transit-delay 3
```

To reset the transit delay to the default value of 1 second, enter:

```
(config-circuit-ip[VLAN6-3.1.2.2])# no ospf transit-delay
```

Showing OSPF Information

Use the **show ospf** command to view OSPF information on the CSS. This command is available in all modes. This section includes the following topics:

- Showing OSPF Area Information
- Showing Global Statistics
- Showing IP Interface Information
- Showing Link-State Databases
- Showing ASE Entries
- Showing the Configured Advertised ASE Routes
- Showing the Redistribution Policy
- Showing Summary Route Configuration Information
- Showing OSPF Neighbors

Showing OSPF Area Information

To show information about OSPF areas, enter:

```
# show ospf areas
```

Table 7-3 describes the fields in the **show ospf areas** command output.

Table 7-3 Field Descriptions for the show ospf areas Command

Field	Description
Area ID	The ID for the area
Type	The area type:Transit or Stub
SPF Runs	The number of times the area calculated the SPF
Area Border Routers	The number of ABRs, including the CSS
AS Boundary Routers	The number of ASBRs, including the CSS, if applicable

Table 7-3 Field Descriptions for the show ospf areas Command (continued)

Field	Description
LSAs	The number of link-state advertisements in the database
Summaries	The capability of summarized LSAs in the stub area, if applicable

Showing Global Statistics

To show OSPF global statistics, enter:

```
# show ospf global
```

Table 7-4 describes the fields in the **show ospf global** command output.

Table 7-4 Field Descriptions for the show ospf global Command

Field	Description
Router ID	The router ID of the CSS.
Admin Status	The state of OSPF on the CSS: Enabled or Disabled.
Area Border Router	Indicates whether the CSS is an ABR. True indicates the CSS is an ABR; otherwise, the field displays False.
AS Boundary Router	Indicates whether the CSS is an ASBR. True indicates the CSS is an ASBR; otherwise, the field displays False.
External LSAs	The number of external LSAs currently contained in the database.
LSA Sent	The number of LSAs sent by the CSS.
LSA Received	The number of LSAs received by the CSS.

Showing IP Interface Information

To show OSPF interfaces, enter:

```
# show ospf interfaces
```

Table 7-5 describes the fields in the **show ospf interfaces** command output.

Table 7-5 Field Descriptions for show ospf interfaces Command

Field	Description
IP Address	The IP address for the OSPF IP interface
Admin State	Administrative state of OSPF on the interface, as affected by the IP interface **ospf enable** command
Area	The area assigned to the interface
Type	The OSPF interface type; always broadcast

Table 7-5 Field Descriptions for show ospf interfaces Command (continued)

Field	Description
State	The functional level of an interface. The state determines whether full adjacencies are allowed to form over the interface. The states include: • **Down** - The initial interface state. In this state, the lower-level protocols indicate the interface is unusable. No protocol traffic is sent or received on the interface. • **Waiting** - The router is trying to determine the identity of the (backup) designated router for the network. To determine the router identify, the router monitors the hello packets it receives. The router is not allowed to elect a backup designated router nor a designated router until it transitions out of the Waiting state. • **DR Other** - The interface is on a network on which another router has been selected to be the designated router. In this state, the router itself has not been selected as the backup designated router. The router forms adjacencies to both the designated router and the backup designated router. • **Backup** - The router itself is the backup designated router on the attached network. The router is the designated router when the present designated router fails. The router establishes adjacencies to all other routers attached to the network. The backup designated router performs slightly different functions during the flooding procedure, as compared to the designated router. • **DR** - The router itself is the designated router on the attached network. Adjacencies are established to all other routers attached to the network. The router must also originate a network LSA for the network node. The network LSA contains links to all routers, including the designated router itself, attached to the network.

Table 7-5 Field Descriptions for show ospf interfaces Command (continued)

Field	Description
Priority	The priority assigned to the interface advertised in the hello packets. When two routers attached to a network both attempt to become the designated router, the router with the highest priority takes precedence. A router whose priority is set to 0 is ineligible to become the designated router on the attached network.
DR	The IP interface address of the designated router selected for the attached network. The designated router is selected on broadcast networks by the hello protocol. Two pieces of identification are kept for the designated router: the Router ID and the IP interface address on the network. The designated router advertises the link state for the network. This network LSA is labeled with the designated router's IP address. The designated router is initialized to 0.0.0.0, which indicates the lack of a designated router.
BR	The backup designated router selected for the attached network. The backup designated router is also selected on all broadcast networks by the hello protocol. All routers on the attached network become adjacent to both the designated router and the backup designated router. The backup designated router becomes the designated router when the current designated router fails. The backup designated router is initialized to 0.0.0.0, indicating the lack of a backup designated router.
Hello	The length of time, in seconds, between the hello packets that the router sends on the interface. This interval is advertised in hello packets sent out on this interface.
Dead	The number of seconds before the router's neighbors declare that the router is down, and when they stop receiving the router's hello packets. This interval is advertised in hello packets sent out on this interface.

Table 7-5 Field Descriptions for show ospf interfaces Command (continued)

Field	Description
Transit Delay	The number of seconds to transmit a Link State Update packet over an interface. LSAs contained in the Link State Update packet have their age incremented by this amount before transmission. This value should take into account transmission and propagation delays; the value must be greater than zero.
Retransmit	The number of seconds between LSA retransmissions for adjacencies belonging to an interface. Also, the interval is used when retransmitting Database Description and Link State Request packets.
Cost	The cost of sending a data packet on the interface, expressed in the link-state metric. The cost of sending a packet is advertised as the link cost for the interface in the router LSA. The cost of an interface must be greater than zero.

Showing Link-State Databases

You can show the entire OSPF link-state database (LSDB) or its specific entry types with the **show ospf lsdb** command. The options for the **show ospf lsdb** command are as follows:

- **show ospf lsdb router** - Displays router LSAs that describe the states of the router interfaces
- **show ospf lsdb network** - Displays network LSAs that describe the set of routers attached to the networkd
- **show ospf lsdb external** - Displays AS-external LSAs that describe routes to destinations external to the AS
- **show ospf lsdb summary** - Displays summary LSAs that describe summarized routes to the network
- **show ospf lsdb asbr_summ** - Displays summary LSAs that describe routes to AS boundary routers

To show the entire database, enter:

```
# show ospf lsdb
```

Table 7-6 describes the fields in the **show ospf lsdb** command output.

Table 7-6 Field Descriptions for the show ospf lsdb Command

Field	Description
Area	The ID for the area.
Type	The link-state type. The types are as follows: • ASB-Summary for summary LSAs originated by ABRs. The LSAs describe routes to ASBRs. • ASE for AS-external LSAs that describe routes to destinations external to the autonomous system. • Network for the network LSAs that describe the set of routers attached to the network. • Router for router LSAs that describe the collected states of the router interfaces. • Summary-Net for summary LSAs originated by ABRs. The LSAs describe routes to networks.
Link State ID	This field identifies the piece of the routing domain that is being described by the LSA. Depending on the link-state type, the Link State ID has following values: • For the ASB-Summary type, the ID is the router ID of the ASBR. • For the ASE type, the ID is the destination network IP address. • For Network type, the ID is the IP interface address of the network designated router. • For Router type, the ID is the originating router’s Router ID. • For Summary-Net type, the ID is the destination network IP address.

Table 7-6 Field Descriptions for the show ospf lsdb Command (continued)

Field	Description
ADV Router	This field specifies the OSPF Router ID of the LSA originator, as follows: • ASB-Summary LSAs, the originators are the ABRs • AS-external LSAs, the originators are ASBRs • Network LSAs, the originators are network-designated routers • Router LSAs, this field is identical to the Link State ID field • Summary LSAs, the originators are the ABRs
Age	The age of the LSA, in seconds. The age is set to 0 when the LSA is originated.
Sequence	A signed 32-bit integer to detect old and duplicate LSAs. The space of sequence numbers is linearly ordered. The larger the sequence number (when compared as signed 32-bit integers), the more recent the LSA. The sequence number 0x80000000 is reserved and unused.
Checksum	The checksum of the complete contents of the LSA, excluding the age field. The age field is excluded to allow the LSA age to increment without updating the checksum. The checksum is used to detect data corruption of an LSA. This corruption can occur while an LSA is being flooded, or while an LSA is being held in a router’s memory. The LSA checksum field cannot take on the value of zero; the occurrence of this value is a checksum failure.

Showing ASE Entries

To show AS-external (ASE) entries in the LSDB, enter:

```
# show ospf ase
```

To find specific entries, pipe the output through the **grep** command. For example: **show ospf ase|grep 10.10.10.0**

Table 7-7 describes the fields in the **show ospf ase** command output.

Table 7-7 Field Descriptions for the show ospf ase Command

Field	Description
Link State ID	The network destination for the advertisement
Router ID	The advertising router
Age	The age, in seconds, of the ASE LSA
T	The ASE type of the route; 1 for ASE Type1 or 2 for ASE Type2
Tag	The tag for the route
Metric	The network cost for the route
FwdAddr	The external destination (forwarding address) for the packets

Showing the Configured Advertised ASE Routes

To show the configuration of ASE routes into OSPF, enter:

```
# show ospf advertise
```

To show the configuration of ASE routes into OSPF for a specific host, include the IP address or host and the subnet mask. Enter the address in dotted-decimal format (for example, 192.168.11.1) or mnemonic host-name format (for example, myname.mydomain.com). Enter the mask either:

- As a prefix length in CIDR bit-count notation (for example, /24). Do not enter a space to separate the IP address from the prefix length.
- In dotted-decimal notation (for example, 255.255.255.0).

For example:

```
    # show ospf advertise 192.168.11.1/24
```

Table 7-8 describes the fields in the **show ospf advertise** command output.

Table 7-8 Field Descriptions for the show ospf advertise Command

Field	Description
Prefix	The IP address for the route. For the CSS, the prefix is predominately VIP addresses.
Prefix Length	The prefix length for the IP address.
Metric	The network cost for the route. The range is from 1 to 16777215. The default is 1.
Type	The ASE type for the route. By default, the ASE type is ASE type2, which is the external cost to reach the route. ASE type1 combines the external and internal costs.
Tag	The 32-bit tag value to advertise the route. The value is not used by OSPF.

Showing the Redistribution Policy

To show the configured redistribution policy into OSPF, enter:

```
    # show ospf redistribute
```

Table 7-9 describes the fields in the **show ospf redistribute** command output.

Table 7-9 Field Descriptions for the show ospf redistribute Command

Static, RIP, Local, or Firewall Field	Description
Routes Redistribution	Indicates whether the redistribution of static, RIP, local or firewall routes is enabled or disabled. If route redistribution is enabled, the configured metric, type, and tag fields are displayed.
Route Metric (displayed when redistribution is enabled)	The external cost for the route. The cost can range from 1 to 16777215. The default is 1.
Route Type (displayed when redistribution is enabled)	The ASE type, either ASE Type1 or ASE Type2. By default, the type is aseType2. The difference between type1 and type2 is how the cost is calculated. For a type 2 ASE, only the external cost (metric) is used when comparing multiple paths to the same destination. For type1 ASE, the combination of the external cost and the cost to reach the ASBR is used.
Route Tag (displayed when redistribution is enabled)	The 32-bit tag value to advertise the external route. The route tag value is not used by the OSPF protocol itself. It is used to communicate information between AS boundary routers.

Showing Summary Route Configuration Information

To show the summary-route configuration information, enter:

```
# show ospf range
```

Table 7-10 describes the fields in the **show ospf range** command output.

Table 7-10 Field Descriptions for the show ospf range Command

Field	Description
Area ID	The ID for the area.
Lsdb Type	The type of link-state database. For an ABR, the type is summaryLink.
Addr Range Mask Range	The address range for the summary route as specified by the IP address (Addr Range) and mask (Mask Range) pair.
Effect	Displays whether the range is advertised or block.

Showing OSPF Neighbors

To show the OSPF neighbors, enter:

```
# show ospf neighbors
```

Table 7-11 describes the fields in the **show ospf neighbors** command output.

Table 7-11 Field Descriptions for show ospf neighbors Command

Field	Description
Address	The IP address of the neighboring router's interface to the attached network. This address is used as the destination IP address when protocol packets are sent as unicasts along this adjacency. The IP address is also used in router LSAs as the Link ID for the attached network if the neighboring router is selected to be the designated router. The CSS learns the neighbor IP address when it receives hello packets from the neighbor.
Neighbor ID	The OSPF Router ID of the neighboring router. The CSS learns the Neighbor ID when it receives hello packets from the neighbor.
Prio	The router priority of the neighboring router. Contained in the neighbor's hello packets, this value is used by OSPF to select the designated router for the attached network.

Table 7-11 Field Descriptions for show ospf neighbors Command (continued)

Field	Description
State/Dr	The state of a conversation being held with a neighboring router. The following states are listed in order of their progression. • Down - The initial state of a neighbor conversation. The Down state indicates that the CSS has received no recent information from the neighbor. • Init - In this state, the CSS has seen a hello packet from the neighbor. However, the CSS has not established bidirectional communication with the neighbor (the router itself did not appear in the neighbor's hello packet). All neighbors in this state (or higher) are listed in the hello packets sent from the associated interface. • 2-Way - In this state, communication between the two routers is bidirectional. The designated router is selected from the set of neighbors in state 2-Way (or greater). • ExStart - This is the first step to create an adjacency between the two neighboring routers. The goal is to decide which router is the master, and to determine the initial Database Description (DD) sequence number. Neighbor conversations in this state (or greater) are called adjacencies.

Table 7-11 Field Descriptions for show ospf neighbors Command (continued)

Field	Description
State/Dr (cont.)	• Exchange - In this state, the CSS sends DD packets to the neighbor to describe its entire link-state database. Each DD packet has a DD sequence number and is explicitly acknowledged. Only one DD packet is allowed to be outstanding at any one time. In this state, the CSS may also send Link State Request packets, requesting the neighbor's more recent LSAs. All adjacencies in Exchange state (or greater) are used by the flooding procedure. In fact, these adjacencies are fully capable of transmitting and receiving all types of OSPF routing protocol packets. • Loading - In this state, the CSS sends Link State Request packets to the neighbor, requesting the more recent LSAs that have been discovered (but not yet received) in the Exchange state. • Full - In this state, the neighboring routers are fully adjacent. These adjacencies will now appear in router LSAs and network LSAs.
Type	Always dynamic.
Rxmt_Q	The number of LSAs to retransmit to the neighbors.

OSPF Configuration in a Startup-Config File

The following example shows an OSPF configuration in a startup-config file.

```
!*************************** GLOBAL ***************************
    ospf router-id 121.23.21.1
    ospf enable
    ospf area 1.1.1.1
    ospf as-boundary
    ospf advertise 192.168.4.15 255.255.255.0
    ospf redistribute static
!************************* INTERFACE *************************
interface ethernet-10
    bridge vlan 6
!************************** CIRCUIT **************************
circuit VLAN6
ip address 192.168.2.2 255.255.255.0
    ospf
    ospf area 1.1.1.1
    ospf priority 0
    ospf enable
```

Where to Go Next

Chapter 8, Using the CSS Logging Features describes how to enable logging, set up the log buffer, and determine where to send the activity information. This chapter also provides information on interpreting sys.log messages and a description of frequently queried messages.

CHAPTER 8

Using the CSS Logging Features

This chapter describes how to enable logging, set up the log buffer, determine where to send the activity information, and display and interpret log messages. Information in this chapter applies to all CSS models, except where noted.

This chapter contains the following major sections:

- Logging Overview
- Specifying Logging Buffer Size
- Configuring Logging for a Subsystem
- Specifying a Log File Destination
- Logging CLI Commands
- Showing Log Files
- Copying Log Files to an FTP or TFTP Server
- Interpreting sys.log Log Messages
- Interpreting Undeliverable Messages
- Frequently Queried Log Messages

Logging Overview

The CSS generates log messages to assist you with debugging and monitoring operations. By default, the CSS saves boot and subsystem event log messages to log files on its hard or Flash disk. The content of these files is recorded in ASCII text. You can also configure the CSS to send log messages to an active CSS session, e-mail address, or another host system.

The maximum size of a log file is 50 MB for hard disk-based systems, and 10 MB for Flash disk-based systems.

The boot log messages are the result of the boot process. The CSS saves these messages in the boot.log file.

The subsystem log messages are subsystem events that occur during the operation of the CSS. The CSS saves these messages in the sys.log file, created when the first loggable subsystem event occurs. The CSS determines which subsystem messages to log by its configured logging level. By default, the CSS logs events on all subsystems with a level of warning. The warning level designates that the CSS logs fatal, alert, critical, error, and warning messages for the subsystem.

You have the option to log subsystem messages at a different level than the default warning level. The level you specify instructs the CSS to log subsystem activity that occurs at that level and the activity greater than that level.

In addition to informational messages, the CSS also logs notice, warning, error, critical, alert, and fatal messages.

You can display or copy a log file using the **show log** or **copy log** command, respectively. For details on the **show log** command, see the "Showing Log Files" section. For details on the **copy log** command, see the "Copying Log Files to an FTP or TFTP Server" section. You need SuperUser privileges to use the **show log** command.

The CSS provides logging capabilities for debugging and system monitoring by generating the log files described in Table 8-1.

Table 8-1 CSS Log File Descriptions

	Log File Destination		
Log File	**Default Location**	**Alternate Location**	**Records**
boot.log	Hard disk and console or Flash disk and console	None	Results of the boot process.
boot.bak	Hard disk and console or Flash disk and console	None	Backup of a boot log file. Each time you reboot the CSS, the software renames the current boot log file to boot.log.prev and starts a new boot log file. The CSS overwrites an existing backup boot log file when a boot log file is renamed.
sys.log	Hard disk or Flash disk	Console syslogd VTY1 VTY2	Log information for user-defined subsystem or CLI commands. By default, logging is enabled and logs subsystem **all** with level **warning**. The CSS creates sys.log to record this log information.

Table 8-1 CSS Log File Descriptions (continued)

	Log File Destination		
Log File	**Default Location**	**Alternate Location**	**Records**
sys.log.prev	Hard disk or Flash disk	Console syslogd VTY1 VTY2	Backup of a system log file. When a system log file reaches its maximum size (50 MB, for a hard disk-based CSS; 10 MB, for a Flash disk-based CSS), the software renames the system log file to sys.log.prev and starts a new system log file. The CSS overwrites an existing backup system log file when a system log file is renamed. When you reboot a CSS, the software continues to use the existing system log file until the file reaches its maximum size.

CSS Logging Quick Start Table

If you are familiar with the CSS logging functions, see Table 8-2 for the commands and command options required to configure and enable logging.

You configure all logging commands from configuration mode except for the clear log command. The clear log command is available only in SuperUser mode at the root prompt (#).

Table 8-2 Configuring and Enabling Logging

Step	Logging Option	Example
1. Specify the disk buffer size.	*size* - Size of the disk buffer (0 to 64000)	**logging buffer 1000**
2. Select a CSS subsystem and determine which type of activity to log (default **all**) and level (default **warning**).	**subsystem** - Valid subsystems: **acl**, **all**, **app**, **boomerang**, **buffer**, **cdp**, **chassis**, **circuit**, **csdpeer**, **dhcp**, **dql**, **fac**, **flowagent**, **flowmgr**, **fp-driver**, **hfg**, **ipv4**, **keepalive**, **netman**, **netmgr**, **nql**, **ospf**, **pcm**, **portmapper**, **proximity**, **publish**, **radius**, **redundancy**, **replicate**, **rip**, **security**, **ssl-accel**, **slr**, **sntp**, **syssoft**, **urql**, **vlanmgr**, **vpm**, **vrrp**, **wcc** **level** - Valid levels: **fatal-0, alert-1**, **critical-2**, **error-3**, **warning-4**, **notice-5**, **info-6, debug-7**	**logging subsystem rip level alert-1**
3. Specify the destination (disk, host, line) where you wish to log subsystem activity.	**disk** *filename* - New or existing filename in the log directory	**logging disk stubs**
	host *ip* or *host* - IP address of the syslog daemon on the host or a host name	**logging host 192.168.11.3** **logging host myhost.domain.com**
	log *line* - CSS active session	**logging line vty1**

Table 8-2 Configuring and Enabling Logging (continued)

Step	Logging Option	Example
4. Optionally, enable the CSS to send log messages to an e-mail address and specify a level.	**sendmail** *email address* of mail recipient *IP address* or *hostname* of SMTP host **level** - Valid levels for the CSS: **fatal-0**, **alert-1**, **critical-2**, **error-3**, **warning-4**, **notice-5**, **info-6**, **debug-7**	**logging sendmail us@arrowpoint.com 172.16.6.58 critical-2**
5. Show the log file.	*filename* - Log file to display	**show log stubs**

Specifying Logging Buffer Size

The logging buffer size is the amount of information the CSS buffers in memory before outputting the information to disk. The larger you configure the buffer size, the less frequently the CSS outputs the contents to disk. Specifying a buffer size is required only if you specify logging to disk as the log file destination.

To set the disk buffering size, use the **logging buffer** command. Specify the buffer size from 0 to 64000 bytes. The default is 0, where the CSS sends the logging output directly to the log file.

To set the buffer size to 1000 bytes, enter:

```
(config)# logging buffer 1000
```

To send the logging output directly to the log file, enter:

```
(config)# no logging buffer
```

Configuring Logging for a Subsystem

This section describes how to select a CSS subsystem and log activity for the subsystem. This section includes the following topics:

- Enabling and Disabling Logging for a Subsystem
- Configuring a Log Message for a Subsystem at a Logging Level
- Logging ACL Activity
- Sending Log Messages to an E-Mail Address

Enabling and Disabling Logging for a Subsystem

Use the **logging subsystem** command to select a CSS subsystem and determine which type of activity to log. The level you specify instructs the CSS to log subsystem activity that occurs at that level and the activity greater than that level. For example, if you wish to log informational messages (info-6), the CSS also logs notice, warning, error, critical, alert, and fatal error levels.

To reset logging for a subsystem to the default logging level (warning-4), enter the **no** version of the logging command. For example:

```
(config)# no logging subsystem redundancy
```

The following example enables logging for the chassis subsystem with a critical-2 error level. The CSS logs all critical, alert, and fatal errors for the chassis.

```
(config)# logging subsystem chassis level critical-2
```

Table 8-3 defines the CSS subsystems for which you can enable logging.

Table 8-3 Logging Subsystems

Subsystem	Definition
acl	Access control list (ACL)
all (default)	All CSS subsystems
app	Application Peering Protocol (APP)
boomerang	DNS Content Routing Agent (CRA)
buffer	Buffer manager

Table 8-3 Logging Subsystems (continued)

Subsystem	Definition
cdp	Cisco Discovery Protocol (CDP)
chassis	Chassis manager
circuit	Circuit manager
csdpeer	Content Server Database (CSD) peer
dhcp	Dynamic Host Configuration Protocol (DHCP)
dql	Domain Qualifier List (DQL)
fac	Flow Admission Control (FAC)
flowagent	Flow agent
flowmgr	Flow manager subsystem
fp-driver	Fathpath driver
hfg	Header Field Group (HFG)
ipv4	Internet Protocol version 4 (IPv4)
keepalive	Keepalive
natmgr	NAT manager
netman	Network management
nql	Network Qualifier List (NQL)
ospf	Open Shortest Path First (OSPF) protocol
pcm	Proximity CAPP Messaging (PCM)
portmapper	Port mapper
proximity	Proximity
publish	Publish
radius	Remote Authentication Dial-In User Service (RADIUS)
redundancy	CSS redundancy
replicate	Content replication
rip	Routing Information Protocol (RIP)
security	Security manager

Table 8-3 Logging Subsystems (continued)

Subsystem	Definition
ssl-accel	Secure Socket Layer (SSL) Acceleration
slr	Session Level Redundancy
sntp	Simple Network Time Protocol (SNTP)
syssoft	System software
urql	Uniform Resource Locator Qualifier List (URQL)
vlanmgr	VLAN manager
vpm	Virtual pipe manager
vrrp	Virtual Router Redundancy Protocol
wcc	Web conversation control

Table 8-4 defines the logging levels you can set for the specified CSS subsystem. The logging levels are listed in order of severity, with a fatal-0 level being the most severe errors and an info-6 level being the least severe error.

Table 8-4 Subsystem Logging Levels

Level	Definition
fatal-0	Fatal errors only.
alert-1	Alert errors, including fatal errors.
critical-2	Critical errors, including alert and fatal errors. The following trap events log at the critical level: link down, cold start, warm start, service down, service suspended.
error-3	General errors, including critical, alert, and fatal errors.
warning-4 (default)	Warning messages, including all lower levels (error, critical, alert, and fatal.
notice-5	Notice messages, including all trap events (except for events logged at critical) and all lower levels except for info and debug.

Table 8-4 Subsystem Logging Levels (continued)

Level	Definition
info-6	Informational messages, including all lower levels except for debug.
debug-7	Debug messages, including all other error levels. The debug-7 log level may degrade the performance of the CSS. When you enter this option, the CSS prompts you with the following message: `Logging at the debug level may degrade the CSS performance. Continue, [y/n]:` Enter **y** to verify that you want to set the log level to debug-7. Enter **n** to cancel the executing of the debug-7 log level.

Configuring a Log Message for a Subsystem at a Logging Level

Use the **cliLogMessage subsystem** command to define a log message for a subsystem at a particular logging level. The syntax for this global configuration mode command is:

cliLogMessage subsystem *name* “*message*” **level** *level*

The variables and options are as follows:

- *name* - The name of a CSS subsystem. Enter one of the subsystem names, as shown in Table 8-3. To see a list of subsystems, enter:

  ```
  cliLogMessage subsystem ?
  ```

- **level** *level* - The log level for the message. Enter one of the levels, from 0 to 7, as shown in Table 8-4. To see a list of levels, enter:

  ```
  cliLogMessage subsystem name “message” level ?
  ```

Logging ACL Activity

When you configure the CSS to log ACL activity, the CSS logs the event of the packet matching the clause and ACL. The CSS sends log information to the location you specified in the **logging** command.

Before you configure logging for a specific ACL clause, ensure global ACL logging is enabled. To globally enable ACL logging, use the **logging subsystem acl level debug-7** command in configuration mode.

To configure logging for an ACL clause:

1. Enter the ACL mode for which you want to enable logging.

   ```
   (config)# acl 7
   (config-acl[7])#
   ```

2. Enable logging for:

   - A new clause, by entering the **log** option at the end of the clause. For example:

     ```
     (config-acl[7])# clause 1 deny udp any eq 3 destination any eq
     3 log
     ```

   - An existing clause, by using the **clause log enable** command:

     ```
     (config-acl[7])# clause 1 log enable
     ```

To disable ACL logging for a specific clause, enter:

```
(config-acl[7])#) clause 1 log disable
```

To globally disable logging for all ACL clauses, enter:

```
(config)# no logging subsystem acl
```

Sending Log Messages to an E-Mail Address

Use the **logging sendmail** command to send the log activity of a subsystem to an e-mail address. The syntax for this global configuration mode command is:

logging sendmail *email_address ip_address level* {*domain*}

The variables are as follows:

- *email_address* - The e-mail address for the recipient. Enter the e-mail address as an unquoted text string with a length of 1 to 30 characters.
- *IP_address* - The IP address for the SMTP host. Enter the IP address in dotted-decimal notation (for example, 192.168.11.1).
- *level* - The type of information to log. The valid levels are defined in Table 8-4.
- *domain* - (Optional) The domain name for the SMTP host. Enter an unquoted text string with a maximum length of 64 characters (for example, arrowpoint.com). Do not insert an @ sign before the domain name. The CSS automatically prepends the @ sign to the domain name.

To turn off logging to an e-mail address, enter:

```
(config)# no logging sendmail email_address
```

Specifying a Log File Destination

To specify a destination where the CSS logs subsystem activity, use the **logging** command. You can specify the following locations for log files:

- **disk** *filename* - New or existing filename in the disk log directory
- **host** *ip* or *host* - IP address of the syslog daemon on the host or a host name
- **log** *line* - CSS active session

Logging to a CSS disk causes the performance of the CSS to degrade.If logging requires frequent writes to disk (that is, several hundred log messages per day), the most reliable configuration is to log to a hard disk and store all other system files on a Flash disk. Although Flash disks generally provide the most reliable way to store information over time, hard disks endure frequent writes to disk better than the Flash disks currently available.

To prevent excessive writes to the CSS disk, consider disabling logging to the sys.log file on disk (see the "Disabling Logging to the sys.log File on the Disk" section). You can continue sending CSS log information to the sys.log file at an alternate location. To do this, use either the **logging host** command to send log information to a syslog daemon on a host system (see the "Specifying a Host for a Log File Destination" section) or the **logging line** command to send (but not save) log information to an active CSS line (see the "Specifying a Line for a Log File Destination" section).

This section includes the following procedures:

- Specifying a Log File on the Disk
- Disabling Logging to the sys.log File on the Disk
- Specifying a Host for a Log File Destination
- Specifying a Line for a Log File Destination

Specifying a Log File on the Disk

Use the **logging disk** command to send log information to a specific file on the CSS disk. Specify a log filename. Enter a text string from 0 to 32 characters. The filename can be new or an existing name.

For example:

```
(config)# logging disk stubs
```

When you specify the **logging disk** command, the CSS:

- Stops writing default log information to the sys.log file
- Creates the filename you specify in the disk log directory
- Sends subsystem and level information to the log file specified

You can have only one active log file on the disk at a time. If you wish to send subsystem information to a different log file on the disk, reenter the logging disk command with a different filename.

⚠ **Caution** Logging to a CSS disk causes the performance of the CSS to degrade.

To stop logging to the specified file and reenable logging to the sys.log file on the CSS, enter:

```
(config)# no logging disk
```

Disabling Logging to the sys.log File on the Disk

Use the **logging to-disk** command to disable logging to the sys.log file on the CSS disk (hard or Flash). Disabling logging to the sys.log file is useful when you want to prevent excessive writes to the CSS disk (for example, to the Flash disk) or to increase the performance of the CSS.

You can continue sending CSS log information to the sys.log file at an alternate location. To send log information to an alternate location, use either the **logging host** command to send log information to a syslog daemon on a host system (see the "Specifying a Host for a Log File Destination" section) or the **logging line** command to send (but not save) log information to an active CSS line (see the "Specifying a Line for a Log File Destination" section).

The options for the **logging to-disk** command are as follows:

- **logging to-disk disable** - Disables writing default log information to the CSS sys.log file on the CSS disk. The **logging to-disk disable** command affects the sys.log file only, and does not affect a disk log file specified through the **logging disk** command. You can still use the **logging disk** *filename* command to send log information to a specific filename on the CSS disk. To disable all logging to the CSS disk, first enter the **no logging disk** command and then enter the **logging to-disk disable** command. When you enter the **no logging disk** command, the CSS does not reenable logging to the sys.log file. You must specify the **logging to-disk enable** command to reactivate the sys.log file.
- **logging to-disk enable** - Resets logging back to disk and resumes writing default log information to the CSS sys.log file.

You are prompted to reboot the CSS after issuing the **logging to-disk disable** or **logging to-disk enable** commands for the command to take effect.

To disable logging to the CSS sys.log file on the CSS disk (Flash disk or hard disk), enter:

```
(config)# logging to-disk disable
```

To resume logging back to the CSS disk, enter:

```
(config)# logging to-disk enable
```

Specifying a Host for a Log File Destination

To send CSS log information to a syslog daemon running on the host system, use the **logging host** command. The syslog daemon receives and displays the CSS log messages on the host system.

The syntax for this configuration mode command is:

logging host *ip_or_host* **facility** *number* **log-level** *number*

The options and variables for this command are as follows:

- *ip_or_host* - Specifies the address of a syslog daemon on the host. Enter the IP address in dotted-decimal notation (for example, 192.168.11.1) or the mnemonic host name (for example, myhost.mydomain.com).
- **facility** *number* - Specifies the syslog daemon facility level. Facilities are considered service areas, such as printing, e-mail, or network. Enter a number from 0 to 7. For more information on the syslog daemon and facility levels, refer to your syslog daemon documentation.
- **log-level** *number* - Specifies the level of the CSS subsystem log messages to be sent to the syslog daemon on the host. The valid log levels for the CSS include: fatal-0, alert-1, critical-2, error-3, warning-4 (default), notice-5, info-6, debug-7. The logging levels are listed in order of severity, with a fatal-0 level being the most severe error and an info-6 level being the least severe error. Refer to Table 8-4 for a definition of the different logging levels.

  The **logging host log-level** *number* must be equal to or less than the log level you configure for the **logging subsystem** command (see the "Configuring Logging for a Subsystem" section). If the log-level value is less than the logging subsystem level, the CSS only sends the message level specified in the **log-level** option. If the log-level is greater than the logging subsystem level, the CSS only sends the level of messages specified in the **logging subsystem** command.

The CSS continues to send log information to the sys.log file on the CSS disk (hard or Flash disk) when the **logging host** command is entered. To disable logging to the sys.log file on the CSS disk, use the **logging to-disk disable** command (see the "Disabling Logging to the sys.log File on the Disk" section).

For example, to send log information to a host at IP address 192.168.11.1 with a facility level of 3 and a log-level of error-3:

```
(config)# logging host 192.168.11.1 facility 3 log-level error-3
```

To turn off logging to a host, enter:

```
(config)# no logging host
```

Specifying a Line for a Log File Destination

To send log information to an active CSS session, use the **logging line** command and specify a valid log line on the CSS. Enter the line as a case-sensitive text string with a maximum of 32 characters.

The CSS continues to send log information to the sys.log file on the CSS disk (hard or Flash disk) even when the **logging line** command is entered. To disable logging to the sys.log file on the CSS disk, use the **logging to-disk disable** command (see the "Disabling Logging to the sys.log File on the Disk" section).

To display a list of active CSS lines, enter the **logging line** command as shown. The asterisk (*) denotes your current session.

```
(config)# logging line ?

console      Login Name:  Location:local
*vty1        Login Name:  admin Location:10.0.3.35
```

To send subsystem information to your monitor, enter:

```
(config)# logging line vty1
```

To turn off logging, enter the **no logging line** command.

```
(config)# no logging line vty1
```

Logging CLI Commands

When you want to keep track of all CLI commands entered from the CSS, you can log them to the sys.log file. To log the CLI commands:

1. Set the logging level of the netman subsystem to info-6. Enter:

   ```
   (config)# logging subsystem netman info-6
   ```

2. Enable logging of commands through the **logging commands enable** command. This command logs each CLI command to the sys.log file. Enter:

   ```
   (config)# logging commands enable
   ```

To disable logging CLI commands to the sys.log file, enter:

```
(config)# no logging commands
```

Showing Log Files

Use the **show log** command to display the contents in a log or trap log file, a list of all log files, or the state of logging for CSS facilities. You need SuperUser privileges to use the **show log** command.

When you use the **show log** command to send the log activity to your current session, and you want to stop sending log activity, press any key on the terminal or workstation. The **show log** command performs the same function as the **logging line** command. You cannot run these commands at the same time.

This section covers:

- Showing Log Activity
- Showing Log Lists
- Showing the Log State

Showing Log Activity

Use the **show log** command and its options to send the log activity to your current session or to display the contents in a log or trap log file. You need SuperUser privileges to use the **show log** command.

The syntax for the **show log** command is:

show log {*log_filename*|**traplog** {**tail** *lines*} {**line-numbers**}}

The options and variables for this command are as follows:

- *log_filename* - Specifies the name of the log file. Enter an unquoted text string with no spaces. To see a list of log files with their creation dates, enter: **show log ?**
- **traplog** - (Optional) Displays all SNMP traps that have occurred. A trap log file is an ASCII file in the log directory containing generic and enterprise traps. By default, the following events generate level critical-2 messages:
  - Link Up
  - Link Down
  - Cold Start
  - Warm Start
  - Service Down
  - Service Suspended

  All other SNMP traps generate level notice-5 messages.

Note When traps are disabled, the CSS still produces a log message for any event that would normally generate a trap.

- **tail** *lines* - (Optional) Displays the bottom and most recent portion of the log file. The CSS displays the log file, starting from the beginning of the file. The top of the file lists the older messages and the bottom lists the most recent messages. You can specify the number of lines to display (to a maximum of 1000 lines), starting at the end of the log file. Enter a number from 1 to 1000.
- **line-numbers** - (Optional) Includes the line numbers when displaying the contents of the log file.

To send the log activity to your current session, enter:

```
# show log
    Displaying Log events.
    Press any key to abort...
    APR 14 16:28:09 5/1 2398 NETMAN-7: HTTPC:HTTPC_Open:
    ERROR->connect <-1,0> <192.20.1.7> <80>
    APR 14 16:28:15 5/1 2399 NETMAN-7: HTTPC:HTTPC_Open:
    ERROR->connect <-1,0> <192.20.1.7> <80>
    APR 14 16:28:21 5/1 2400 NETMAN-7: HTTPC:HTTPC_Open:
    ERROR->connect <-1,0> <192.20.1.7> <80>
    APR 14 16:28:27 5/1 2401 NETMAN-7: HTTPC:HTTPC_Open:
    ERROR->connect <-1,0> <192.20.1.7> <80>
```

To display information in a specific log file, enter the **show log** command with a valid log filename. For example:

```
# show log stubs
    SEP 22 09:59:18 5/1 918 NETMAN-7: SNMP:SET RSP (3803)
    SEP 22 09:59:53 5/1 919 NETMAN-7: SNMP:SET  (3804)
    SEP 22 09:59:53 5/1 920 NETMAN-7: SNMP:  1
    apLogHostIpAddress.[1.2.3.4] VT_IPADDRESS  <1.2.3.4>
    SEP 22 09:59:53 5/1 921 NETMAN-7: SNMP:  2
    apLogHostIpAddress.[1.2.3.4] VT_IPADDRESS  <1.2.3.4>
```

To view the content of the sys.log file, enter:

```
    (config)# show log sys.log
```

To view the bottom and most recent portion of the file, use the **tail** option with the **show log** command. For example, to view the most recent 500 lines in the sys.log file, enter:

```
    (config)# show log sys.log tail 500
```

Showing Log Lists

Use the **show log-list** command to display a list of all log files. For example:

```
    (config)# show log-list
```

Showing the Log State

Use the **show log-state** command to display the state of logging for CSS facilities. For example:

```
(config)# show log-state
```

Table 8-5 describes the fields in the **show log-state** command output.

Table 8-5 Field Descriptions for the show log-state Command

Field	Description
Subsystems:	
acl	Access Control List (ACL)
app	Application Peering Protocol (APP)
boomerang	DNS Content Routing Agent (CRA)
buffer	Buffer manager
cdp	Cisco Discovery Protocol (CDP)
chassis	Chassis manager
circuit	Circuit manager
csdpeer	Content Server Database (CSD) peer
dhcp	Dynamic Host Configuration Protocol (DHCP)
dql	Domain Qualifier List (DQL)
fac	Flow Admission Control (FAC)
flowagent	Flow agent
flowmgr	Flow manager subsystem
fp-driver	Fathpath driver
hfg	Header Field Group (HFG)
ipv4	Internet Protocol version 4 (IPv4)
keepalive	Keepalive
natmgr	NAT manager
netman	Network management
nql	Network Qualifier List (NQL)

Table 8-5 Field Descriptions for the show log-state Command (continued)

Field	Description
ospf	Open Shortest Path First (OSPF)
pcm	Proximity CAPP Messaging (PCM)
portmapper	Port mapper
proximity	Proximity
publish	Publish
radius	Remote Authentication Dial-In User Service (RADIUS)
redundancy	CSS redundancy
replicate	Content replication
rip	Router Information Protocol (RIP)
security	Security manager
ssl-accel	Secure Socket Layer (SSL) Acceleration
slr	Session Level Redundancy
sntp	Simple Network Time Protocol (SNTP)
syssoft	System software
urql	Uniform Resource Locator Qualifier List (URQL)
vlanmgr	VLAN manager
vpm	Virtual pipe manager
vrrp	Virtual Router Redundancy Protocol
wcc	Web conversation control
acl	Access Control List (ACL)
all (default)	All CSS subsystems
app	Application Peering Protocol (APP)
Levels:	
debug	Log all errors and messages (Verbose)
info	Log informational messages, including errors at the notice level

Table 8-5 Field Descriptions for the show log-state Command (continued)

Field	Description
notice	Log notice messages, including errors at the warning level
warning	Log warning errors (default), including errors at the error level
error	Log errors, including errors at the critical level
critical	Log critical errors, including errors at the alert level
alert	Log alert errors, including errors at the fatal level
fatal	Log fatal errors only (Quiet)
File:	
Filename:	Name of the log file
Current size:	Current size of the log file
Log to Disk	Identifies whether logging to disk (Flash disk or hard disk) is Enabled or Disabled.

Copying Log Files to an FTP or TFTP Server

Use the **copy log** command to copy log files from the CSS to a File Transfer Protocol (FTP) or Trivial File Transfer Protocol (TFTP) server. The **copy log** command is available only in SuperUser mode.

This section includes the following sections:

- Copying Log Files to an FTP Server
- Copying Log Files to a TFTP Server

Copying Log Files to an FTP Server

Use the **copy log ftp** command to copy a log file to an FTP server. Before you copy a log file from the CSS to an FTP server, create an FTP record file containing the FTP server IP address, username, and password. Refer to Chapter 3, Managing the CSS Software for information on configuring an FTP record.

The syntax is:

copy log *logfilename* **ftp** *ftp_record filename*

The options and variables for this command are as follows:

- *logfilename* - Specifies the name of the log file on the CSS. Enter an unquoted text string with no spaces and a maximum of 32 characters. To see a list of log files, enter the **copy log ?** command.
- **ftp** - Copies a log file to an FTP server.
- *ftp_record* - Specifies the name of the FTP record file that contains the FTP server IP address, username, and password. Enter an unquoted text string with no spaces and a maximum of 16 characters. To create an FTP record, see Chapter 3, Managing the CSS Software.
- *filename* - Specifies the name you want to assign to the file on the FTP server. Include the full path to the file. Enter an unquoted text string with no spaces and a maximum of 32 characters.

For example, to copy the *starlog* log file to an FTP server:

```
# copy log starlog ftp ftpserv1 starlogthurs
```

Copying Log Files to a TFTP Server

Use the **copy log tftp** command to copy a log file to a TFTP server.

The syntax is:

copy log *log_filename* [**ftp** *ftp_record*|**tftp** *ip_or_host*] *filename*

The options and variables for this command are as follows:

- *log_filename* - The name of the log file on the CSS. Enter an unquoted text string with no spaces and a maximum of 32 characters. To see a list of log files, enter the **copy log ?** command.
- **tftp** - Copies a log file to a TFTP server.
- *ip_or_host* - The IP address or host name of the TFTP server to receive the file. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1) or in mnemonic host-name format (for example, myhost.mydomain.com). If you wish to use a host name, you must first set up a host table using the **host** command.

- *filename* - The name you want to assign to the file on the TFTP server. Include the full path to the file. Enter an unquoted text string with no spaces and a maximum of 32 characters.

For example, to copy the *starlog* log file to a TFTP server:

```
# copy log starlog tftp tftpserv1 starlogthurs
```

Interpreting sys.log Log Messages

The following example shows a sys.log message. This section describes the parts of a log message using this example.

```
FEB 16 14:01:13 5/1 2453 VLANMGR-7: Transmit sfm STP BPDU on bPort 1,
egressLp 0x1f00 VlanLpSend() ret:0
```

A log message consists of the following components:

- The time stamp indicates when the log message event occurred. In this example, the time stamp is FEB 16 14:01:13.
- The physical interface indicates the *slot/port* (for example, 3/1) where the event occurred in the CSS.
- The counter records the incremental occurrence of each message. The count of this message is 2,453.
- The subsystem name and level is the CSS subsystem assigned to the message and the level of the message. Because this example is a subsystem message, the subsystem is the VLAN Manager and the log level is 7, which is a debug level (VLANMGR-7). See the "Configuring Logging for a Subsystem" section for a list of CSS subsystems.
- The log message indicating the event has occurred. The remaining string in the example is the event that occurred.

  ```
  Transmit sfm STP BPDU on bPort 1, egressLp 0x1f00 VlanLpSend()
  ret:0
  ```

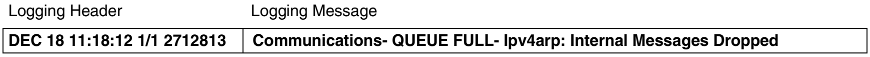
You can define a log message for a subsystem at a particular logging level through the **cliLogMessage subsystem** command. For more information, see the "Configuring a Log Message for a Subsystem at a Logging Level" section.

Interpreting Undeliverable Messages

Undeliverable messages, such as IMM, EVENT, and LOCAL, appear in the log when a queue on the CSS becomes full, overutilized, or needs to state a problem. The CSS supports EVENT and LOCAL messages.

Undeliverable messages consist of a logging header and a logging message, as shown in Figure 8-1.

Figure 8-1   Undeliverable Message Format

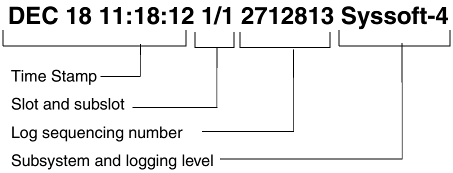


Logging Header	Logging Message
DEC 18 11:18:12 1/1 2712813	Communications- QUEUE FULL- Ipv4arp: Internal Messages Dropped

The logging header (Figure 8-2) contains:

- A time stamp with the date and time
- The CSS slot and subslot number
- A logging sequence counter indicating that log messages were dropped due to excessive logging or CSS processor load
- The subsystem and its log level

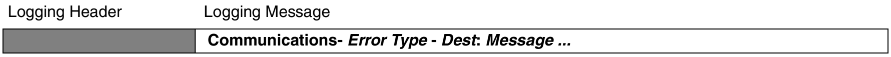
Figure 8-2   Logging Header in a Log Message



```
DEC 18 11:18:12 1/1 2712813 Syssoft-4
```

The logging message that follows the logging header defines the error type (*Error Type*) and the destination (*Dest*), followed by a message (see Figure 8-3).

Figure 8-3 Logging Message



The error type indicates one of the following:

- QUEUE FULL - The receiving queue has no room to accept messages
- QUEUE DELETED - The CSS was trying to place a message in a valid queue, the message was destroyed
- QUEUE INVALID - A destination message queue handle was not a valid object
- QUEUE UNKNOWN - The CSS was trying to determine the destination queue, the lookup failed.

The destination indicates one of the following:

- String decoded name of the destination message queue
- Hexadecimal value if the error type is QUEUE DELETED, QUEUE INVALID, or QUEUE UNKNOWN
- INTERNAL, a LOCAL message passed between the threads on the same processor

The message (*Message...*) in the logging message provides additional information concerning the problem. The log level for the undeliverable message determines how much information is in the logging message and how often the message occurs in the log. You can set the log level to Warning-4, Info-6, or Debug-7.

By default, the log level for undeliverable messages is Warning-4. These messages occur every two seconds per message queue that is experiencing the problem. The message in the logging messages provides only the following information:

```
Internal Messages Dropped.
```

When you change the log level to Info-6, the undeliverable message still occurs every two seconds per message queue that is experiencing the problem. However, the logging message displays a message similar to the following:

```
Internal Messages dropped 5 times since the previous log for a
total of 21 times since bootup.
```

This message provides additional information such as:

- How many times the internal messages were dropped since the previous logging of the message
- The total number of dropped messages since the CSS bootup

When you require more detailed information, set the log level to Debug-7. An undeliverable message appears in the log each time it occurs. The logging message displays a message similar to the following:

```
Message (IMM:Base Class-SYS_Event, Identifier 0) from 1/1 (other
CSS) failed to reach destination ‘Ipv4Arp’ on 1/1 (this CSS)
```

The fields in this message includes a message type, details based on the message type, source information, and destination information. See Table 8-6 for descriptions of these fields.

```
Message (message_type:details) from Source_information failed to
reach destination Dest on Dest_info
```

Table 8-6 Message Fields in a Log Level Debug-7 Logging Message

Message Fields	Description and possible entries
message type (*message_type*)	IMM - Message passed both as inter-processor or intra-processor messages. LOCAL - Message passed between the threads on the same processor and may be IMM in format. EVENT - Messages to be distributed to a registered set of recipients throughout the entire CSS.

details (*details*)	For a LOCAL message type, there is no detail.
	For an EVENT message type, the details can be one of the following:
	• String decoded name of the event when the event is known:
	`(Event:Ipv4ArpChangeEvent)`
	• Hexadecimal encoded name of the event when the event is out of range:
	`(Event:unknown type-0x00a00005)`
	For an IMM message type, the details include Base Class followed by one of the following:
	• The string decoded name of the base task ID and an identifier that is the decimal instance in the class, for example:
	`(IMM:Base Class-SYS_Event, Identifier 0)`
	• Unknown and the hexadecimal value of the message type field, for example:
	`(IMM:Base Class- Unknown, unknown type-0x00a00005)`

Table 8-6 Message Fields in a Log Level Debug-7 Logging Message

Message Fields	Description and possible entries

Source information (*source_information*)	The origin of the message. The information includes the slot and subslot numbers and whether they are on either "this CSS" or "the other CSS" in an Adaptive Session Redundancy (ASR) configuration. For example:
	`from 1/1 (this CSS)`
	A LOCAL message type also includes information for the local processor context in string format, for example:
	`from 1/1- 'EventAgent' (this CSS)`

Destination (*Dest*) and Destination Information (*Dest_Info*)	The destination is the same destination as the destination at the beginning of the logging message. The destination information is where the message is going. This information includes the slot and subslot numbers as they appears in the logging header. For example: `failed to reach destination Ipv4Arp on 1/1 (this CSS)`

Frequently Queried Log Messages

Table 8-7 lists the frequently queried log messages for the Cisco 11500 series CSS. This table includes information on the possible cause and corrective action, if required. Log messages are divided by logging subsystem, with messages listed alphabetically.

Table 8-7 Cisco 11500 Series CSS Log Messages

Log Message (sys.log: Subsystem Name, Level, and Message)	Cause and Resolution
acl Subsystem	

`ACL-7: ACL match 2:254 Discarding`
`ACL-7: TCP SrcPort: 1043 DestPort: 21`
`ACL-7: Source: 172.20.57.2`
`ACL-7: Dest: 172.20.48.35`
	Incoming traffic matches an ACL statement. The CSS examines, and then drops, the packet.
	The log message appears for a packet that has an ACL statement applied by the flow manager. This log message indicates that load balancing can take place.

`ACL-7: ACL rule match 2:254 Discarding packet, Log Enabled`	Incoming traffic matches an ACL statement. The CSS examines, and then drops, the packet. The log message appears for a packet that has an ACL statement applied by the IPV4 module. This log message indicates that the CSS may not set up flows for the packet (certain source or destination ports do not create a flow). This log message could also be related to issues with ICMP or RIP.

Table 8-7 Cisco 11500 Series CSS Log Messages

Log Message (sys.log: Subsystem Name, Level, and Message)	Cause and Resolution
chassis Subsystem	
`CHMGR: Missing backup power supply.`	The power supply lost AC power from the source. A CSS 11501 and CSS 11503 contains one power supply. The CSS 11506 contains up to three power supplies, but requires two functioning power supplies to guarantee service. If the following message appears first, then you can assume that the problem is with the AC power source, not the power supply. `CHMGR: Cannot locate power supply: PS`*`number`*`.` The *PSnumber* variable indicates which power supply cannot be found or has failed. To determine whether the CSS power supplies are working properly, both LEDs on the front of each power supply should be lit.

Table 8-7 Cisco 11500 Series CSS Log Messages

Log Message (sys.log: Subsystem Name, Level, and Message)	Cause and Resolution
`CHMGR: Cannot locate power supply: PS`*`number`*`.`	The CSS chassis cannot find the power supply. The CSS 11501 and CSS 11503 contain one power supply. The CSS 11506 contains up to three power supplies but requires two functioning power supplies to guarantee service. The PS*number* variable indicates which power supply cannot be found or has failed. If you know that the power source is supplied to the chassis and correctly flowing to it, then the problem may be the power supply. To determine whether the CSS power supplies are working properly, make sure that both LEDs on the front of each power supply are lit.

Table 8-7 Cisco 11500 Series CSS Log Messages

<table>
<tr><th>Log Message (sys.log: Subsystem Name, Level, and Message)</th><th>Cause and Resolution</th></tr>
<tr><td colspan="2"><b>circuit Subsystem</b></td></tr>
<tr><td><code>CIRCUIT-7: Circuit status message for circuit 1023 sent to CE 20202c01 cause code is 7</code></td><td>Codes indicate the status of interfaces within VLANs. The logical port cause and command codes are as follows:
<table>
<tr><th>Cause</th><th>Code</th></tr>
<tr><td>CM_CIRCUIT_CREATED</td><td>1</td></tr>
<tr><td>CM_IP_REGISTER</td><td>2</td></tr>
<tr><td>CM_IP_NOT_REGISTER</td><td>3</td></tr>
<tr><td>CM_IP_MODIFIED</td><td>4</td></tr>
<tr><td>CM_LP_STATE_CHG</td><td>5</td></tr>
<tr><td>CM_CIRCUIT_REMOVED</td><td>6</td></tr>
<tr><td>CM_LP_ADDED</td><td>7</td></tr>
<tr><td>CM_LP_REMOVED</td><td>8</td></tr>
<tr><td>CM_LP_MODIFIED</td><td>9</td></tr>
<tr><td>CM_LP_FAILOVER</td><td>10</td></tr>
<tr><td>CM_CIRCUIT_DOWN</td><td>11</td></tr>
</table>

This log message indicates that a port has been added to a VLAN. This log message can occur when the association to a VLAN changes as the port transitions from an up to a down state.
Use the <b>show circuit</b> command to list the VLANs (refer to Chapter 5, Configuring Interfaces and Circuits). Check the status of the ports of the VLAN or determine whether the VLAN is active.</td></tr>
</table>

Table 8-7 Cisco 11500 Series CSS Log Messages

Log Message (sys.log: Subsystem Name, Level, and Message)	Cause and Resolution
csdpeer Subsystem	
`CSDPEER-7: LR Send list too small !!!`	The number of domain names sent by the peer exceeds the size of the CSS list. You can configure this parameter through the **(config) dns-peer** command (see the *Cisco Content Services Switch Advanced Configuration Guide*). We recommend that you configure the receive and send slots with the same value. The default slot value is 255.
flowmgr Subsystem	
`FLOWMGR-4: Flow manager received an illegal message with code 10`	One of the Ethernet ports received a high number of malformed packets, resulting in an overflow of the fastpath. In this case, the flow manager received a badly formatted control message from the fastpath. This problem may be due to intermittent hardware, which results in the fastpath corrupting the packets, or the problem is related to the fastpath receiving streams of malformed packets and leaking some of those packets to the flow manager. Use the **show ether-errors** command to display information for a port that is experiencing many errors. Try disconnecting the port or changing ports and determine whether the errors stop.
`FLOWMGR-4: Flow manager received an illegal message with code 255`	One of the CSS Ethernet ports encountered a significant number of errors and a few malformed packets reached the flow manager.

Table 8-7 Cisco 11500 Series CSS Log Messages

Log Message (sys.log: Subsystem Name, Level, and Message)	Cause and Resolution
`FLOWMGR-4: Flow manager received an illegal message with code 194`	The flow manager received an illegal message from the fastpath. This log message may occur due to hardware problems or a port receiving an excessive number of malformed packets. Use the **show mibii** or **show ether-errors** command to look for errors on one of the CSS ports.

`FLOWMGR-6: FM_Tcp: Handling generic FMTCP flow Re-Transmit ERROR`
`FLOWMGR-6: FM_ReTransTimeout: Re-Transmit timeout ERROR`
`FLOWMGR-6: FM_Tcp: Handling generic FMTCP flow Re-Transmit ERROR`
`FLOWMGR-6: FM_ReTransTimeout: Re-Transmit timeout ERROR`
`FLOWMGR-6: FM_Tcp: Handling generic FMTCP flow Re-Transmit ERROR`
	If the CSS handles a content request for a Layer 5 rule that spans more than three TCP packets, after the CSS decides on the server to use, it sends the TCP packets in TCP slow start form. Here is an example of a five-TCP segment content request:
	`Segment 1 -->`
	`Segment 2 -->`
	`(wait for an ACK)`
	`<--- ACK`
	`Segment3 ->`
	`Segment4 ->`
	`Segment5 ->`
	`<-- Content`
	If the CSS does not receive an ACK from the server within three seconds, it will refuse any remaining packets and terminate the connection with a reset. At that point, the log message is generated.

Table 8-7 Cisco 11500 Series CSS Log Messages

Log Message (sys.log: Subsystem Name, Level, and Message)	Cause and Resolution
`FLOWMGR-6: \n FM_UtilGenericTcpFlowReject: Handling Generic Flow REJECT`	The CSS rejects a connection, either due to issues with the client side or the server side of the connection. When this log message occurs as a result of the client side connection, the issue could be due to a content request that spans more than the configured maximum (default is 6). Other reasons could include: • A delayed ACK could not be sent to the client (at 200ms) • A delayed ACK is sent to the client, and the client responds back with a TCP SYN/FIN/RST handshake sequence. • The client side closes down unexpectedly. When this log message occurs as a result of the server side connection, the issue could be due to the CSS sending the spanned content request to the server and did not get an acknowledgement from the server or received an unexpected response (for example, due to the flow being torn down to the client side). If this message appears frequently in the log, contact Cisco Systems TAC.
`FLOWMGR-7: Allocation for a vector-loaded flow, where theFlow = 840ef5b0`	This is an informational message. No further action is required.

Table 8-7 Cisco 11500 Series CSS Log Messages

Log Message (sys.log: Subsystem Name, Level, and Message)	Cause and Resolution

`FLOWMGR-7: Exceeded outflow SYN count`	For a Layer 5 rule, the CSS is trying to establish a connection with the backend server. The CSS sent four SYNs to the backend server and did not get a response.
	For the CSS to establish a connection with the backend server, the CSS must receive the following TCP/IP handshake:
	`SYN->`
	`<-SYN/ACK`
	`ACK->`
	`GET->`
	After receiving the GET message, the CSS opens the backend connection. At that point, the log message is generated.
	When several of these log messages occur, there might be a malfunctioning server. The server problem could be from keepalives, or from regular TCP HTTP traffic. Make sure the port 80 sockets are not full on the servers.

fp-driver	
`FP_DRV-4: PrismImmFastPath::Send: Could not allocate an MCID. Remote message send aborted.`	The CSS MIPS processor attempts to send a group message, but the processor cannot obtain an multicast ID (MCID) from the Multicast ID Module (MID). The MID keeps track of the reference count on a buffer when the CSS sends a packet to multiple locations. The fast path uses an MCID to reference count the buffer with a contained packet that is being flooded to all ports in a VLAN.

Table 8-7 Cisco 11500 Series CSS Log Messages

Log Message (sys.log: Subsystem Name, Level, and Message)	Cause and Resolution
`FP_DRV-4: PrismImmFastPath::Send: Could not allocate an MCID. Remote message send aborted.` (continued from previous page)	For MCIDs to be depleted in the CSS, there must be 1024 reference counted packets queued up in some combination of hardware queues and software queues. To fill up hardware queues, the CSS must receive a large number of packets, which it then chooses to flood out all ports. In the case of software queues, it might be possible for some task on the MIPS processor to deplete the MCID pool by sending a large number of messages to a group that contains both local and remote members. If the local member has a very large queue, the queue could fill up before running the recipient task, processing the messages, and freeing the buffers. Of the two potential causes, the most likely causes is the CSS receiving a large number of packets which it must flood out all ports. This log message typically occurs when the CSS loses a specific route and forwards the flow out the default gateway. The default gateway then forwards the flow back to the CSS because the routing table lists the CSS as the next hop. As a result, the packet is continuously routed out the default gateway and back to the CSS.

Table 8-7 Cisco 11500 Series CSS Log Messages

Log Message (sys.log: Subsystem Name, Level, and Message)	Cause and Resolution
ipv4 Subsystem	

`IPV4-4:Ipv4IfMgrCctUpdateMsg: IF config for circuit 1015 not found`
`CIRCUIT-4: Error, Circuit 1015 does not exist.`
	You deleted a circuit but the circuit is still referenced by an ACL or to another configuration parameter. Verify the CSS configuration and make the necessary modifications to remove references to the deleted circuit.

`IPV4-4: Ipv4ReceivePacket: out of mbufs`	An *mbuf* is a data structure in BSD UNIX-based IP stacks (such as the VxWorks stack) that is used for buffering. This log message indicates the CSS received a packet that was addressed to a CSS IP address, and when attempting to send the packet up the VxWorks IP stack, the CSS had no remaining buffers. **Note** These buffers are separate from those used for flow setup and forwarding purposes. They are used only when traffic is sent to the CSS itself, (for example, during a Telnet session). If you receive this message, contact Cisco Systems TAC.

Table 8-7 Cisco 11500 Series CSS Log Messages

Log Message (sys.log: Subsystem Name, Level, and Message)	Cause and Resolution
`IPV4-4: Ipv4SfmArpTx: unknown circuit in buffer (2001)`	An ARP TX task is running on the SFM, receiving packets from the SFM and transmitting them to the proper egress ports. This message includes the circuit number (2001, in this example). If the circuit number for this ARP was down or inactive while the ARP was still being queued in the SFM, the message would appear in the log. An action caused the circuit to be removed while data for the circuit was still in the buffer. Determine whether all physical interfaces in a circuit VLAN are going up and down, or a configuration change occurred on the VLAN at the time of the message.
`IPV4-4: Ipv4SfmForwRx: bad IP version received (0)`	The IPV4 receive task received a packet and the IP version is displayed in parentheses (). The CSS discards any packet that is not Ipv4 version 4. In this example, the IP version is 0. If you see many of these messages, the problem could be an improperly configured device or a DoS attack.
`IPV4-4: Ipv4SfmForwTx: No VC for buffer (0x00000000)`	The IPV4 transmit task has a buffer to transmit to the switch fabric processor, however the CSS still needs to create the Virtual Circuit to the fastpath. This message typically occurs if the CSS Ethernet port changes state. This is an informational message. No further action is required.

Table 8-7 Cisco 11500 Series CSS Log Messages

Log Message (sys.log: Subsystem Name, Level, and Message)	Cause and Resolution
`IPV4-4 Ipv4SfmForwTx: unknown logical port in buffer <0x05c01f00>`	A link became unavailable while an ARP or other IPV4 packets were in transit. When this occurred, the CSS chose a logical port to transmit from, formatted the packets, and then attempted to transmit the packets. At the point when the CSS attempted to transmit packets, the logical port was no longer available. This is an informational message. No further action is required.
`IPV4-4: Ipv4ApIoctl: unknown command: 1074031872`	This is an informational message. No further action is required.
`IPV4-4: Ipv4SfmForwRx: buffer length (872) less than IP length (1004)`	IP packets have been corrupted and the IP header Total Length value does not match with the actual length of the packet. In this case, the SP receives less total bytes than expected from the IP header length. This message may be related to hardware problems or errors on the line (corrupted packets). Use the **show mibii** or **show ether-errors** command to look for errors on one of the CSS ports.

Table 8-7 Cisco 11500 Series CSS Log Messages

Log Message (sys.log: Subsystem Name, Level, and Message)	Cause and Resolution
`IPV4-4: (RIP) VIP Redundancy callback on unregistered address 0.0.0.0 range 0`	The CSS is running VIP redundancy and is using the global Routing Information Protocol (RIP) to advertise VIPs on the CSS to other routers (configured through the **rip advertise** command). In this case, RIP sends the redundancy manager in the CSS the VIP and the range, and requests to be informed of any changes in the VIP redundancy status. If the redundancy manager monitors a change in VIP redundancy status, it contacts RIP with the VIP address and the range. To ensure the proper advertisement of the VIPs, RIP verifies the VIP address and range. This log message occurs when RIP is unable to find the VIP address received in the callback message from the redundancy manager. Check the IP address specified in the **rip advertise** command and verify that the VIPs are configured properly for VIP and virtual interface redundancy.
`IPV4-0: Ipv4SfmProcessArpFrame: ARP packet with unknown ingress port 0x0fc01f00`	A CSS Ethernet port became unavailable while an ARP packet is in transit from the fastpath to the IPV4 destination. As a result, the ARP packet is dropped. This is an informational message. No further action is required.

Table 8-7 Cisco 11500 Series CSS Log Messages

Log Message (sys.log: Subsystem Name, Level, and Message)	Cause and Resolution
`IPV4-4: Ipv4SfmCmDeleteFlow: -1 response from VccRemoveVc, egress 0x09c01f00`	A CSS Ethernet port became unavailable. As a result, the IPV4 module was unable to delete a Virtual Circuit established through the switch fabric to the fastpath. This is an informational message. No further action is required.

`IPV4-4: Ipv4SfmProcessArpFrame: bad ARP packet received`	The CSS detects that it has received an invalid ARP packet. The following messages can appear in the log to clarify why the CSS logs the receipt of an invalid ARP packet.
	`IPV4-4: ffff53ff01ff ff0077fa6503 0806`
	`IPV4-4: HW type: 0x0000  Proto type: 0x0000`
	`IPV4-4: HLEN 0x00  PLEN 0x00`
	`OPTPA-TSI-CSS1# 0x0000`
	`IPV4-4: Sender HA 000000000000`
	`IPV4-4: Sender IP 0.0.0.0`
	`IPV4-4: Target HA 000000000000`
	`IPV4-4: Target IP 0.0.0.0`
	The first line in the log message identifies the destination MAC address of the packet, the source address of the packet, and the type of IP packet. In the example above, the destination MAC address is ff-ff-53-ff-01-ff and the source MAC address is ff-00-77-e7-65-03. In addition, 0806 equals ETHERTYPE_ARP.
	The second line in the message identifies the hardware type and protocol type. In this example, the CSS logs the messages because both the hardware type and protocol type are **0000**. The CSS views this value as an indication of an invalid packet.

Table 8-7 Cisco 11500 Series CSS Log Messages

Log Message (sys.log: Subsystem Name, Level, and Message)	Cause and Resolution

`IPV4-4: Duplicate IP address detected: 192.168.163.129 00-08-e2-10-38-54`
`IPV4-4: Incoming CE 0x3001f04, incoming (0 based) SLP 0xc`
	A duplicate IP address has been detected by the CSS. Typically, two level 4 IPV4 messages appear to provide assistance in finding the duplicate IP address that was detected by the CSS.
	The first message states that the CSS received a packet with a source IP address that is also configured on the CSS. The message identifies the duplicate source IP address and its corresponding MAC address as an aid to locate the device with the duplicate IP address.
	The second message is intended to assist you in locating the port on the CSS that has received the duplicate IP address. Use the **flow statistics** command to locate the interface on the CSS. The **flow statistics** command should correspond the CE value listed in the log message with a port.

keepalive Subsystem	

`KAL-7: kal_ServiceNotify: kalIndex = 24 kalSvcEvent=3`
`KAL-7: kal_ServiceNotify: kalIndex = 31 kalSvcEvent=4`
`KAL-7: kal_ServiceNotify: kalIndex = 49 kalSvcEvent=5`
	The CSS is configured with HTTP keepalives (HEAD or GET) and the servers transition between states. The service event (kalSvcEvent) values are as follows:
	• 3 = Alive
	• 4 = Dying
	• 5 = Dead
	Check the status of the server.

Table 8-7 Cisco 11500 Series CSS Log Messages

Log Message (sys.log: Subsystem Name, Level, and Message)	Cause and Resolution
netman Subsystem	
`NETMAN-1: TRAP:Authentication:Generated by: 192.168.36.252`	The CSS is configured to transmit SNMP trap messages and a user attempts to access the CSS with an incorrect SNMP community string. In this example, the CSS sends a trap to the configured SNMP trap receiver stating that a client with IP address 192.168.36.252 is trying to access the CSS with an incorrect community string. This log message also appears when a user attempts to access the CSS using SNMP and SNMP is not configured on the CSS. Refer to Chapter 12, Configuring Simple Network Management Protocol (SNMP) for configuration information.
`NETMAN-2: Sshd:do_authenticated:ERROR-> TSM Rejects connection`	Remote access is being initiated to the CSS CLI. If the CSS security manager rejects the log in, the session terminates. The security manager can reject the log in when: • The maximum number of concurrent security manager users had been exceeded (128 concurrent users). • The CSS could not re-register (if you had a session that just ended and the flow cleanup was not performed, and you attempted to re-register too soon). • The CSS ran out of memory and could not allocate a control block.

Table 8-7 Cisco 11500 Series CSS Log Messages

Log Message (sys.log: Subsystem Name, Level, and Message)	Cause and Resolution

`NETMAN-2: Generic:LINK DOWN for 13/1`
`CIRCUIT-6: Port 13/1 is down for circuit VLAN1`
`NETMAN-2: Generic:LINK DOWN for 13/2`
`CIRCUIT-6: Port 13/2 is down for circuit VLAN1`
`NETMAN-2: Generic:LINK DOWN for 13/3`
`CIRCUIT-6: Port 13/3 is down for circuit VLAN1`
`NETMAN-2: Generic:LINK DOWN for 13/4`
`CIRCUIT-6: Port 13/4 is down for circuit VLAN1`
`SYSSOFT-3: ONDM: Timeout downloading image to EPIF 0 from the switch.`
`SYSSOFT-3: ONDM: Timeout downloading image to EPIF 0 from the switch.`
	EPIF 0 belongs to the first four ports on a FEM. This log message usually relates to a problem with the SFM not getting the code to the FEM or the FEM not reading the SFM properly.
	In this case, there is a communications problem between the SP 9/2 and the FEM.
	`JAN  5 00:31:43 arrowpoint1.com 9/2 385390 SYSSOFT-3: ONDM: Timeout downloading image to EPIF 0 from the switch.`
	`JAN  5 00:31:45 arrowpoint1.com 9/2 385407 SYSSOFT-3: ONDM: Timeout downloading image to EPIF 0 from the switch.`
	Reseat the SFM in slot 9, then reseat the FEM in slot 13 that is controlled by the SFM. Cycle power to the CSS.

`NETMAN-2: Enterprise:Service Transition:ServerA -> down`
`NETMAN-5: Enterprise:Service Transition:ServerA -> alive`
	This is an information message when the service has changed state. Check the status of the server based on the keepalive parameters.

`NETMAN-4: SNMPAPI:SNMPAPI_Set:SET failure`	A user on the CLI, connected to the CSS either through the console or by Telnet, has entered an incorrect command. A Telnet or console session displays this message, stating that the command was incorrect.
`NETMAN-5: Enterprise: Login Failure:vty2 10.6.3.171 Mandy`	SNMP Enterprise login failure traps are enabled and an invalid username and password have been entered. Refer to Chapter 12, Configuring Simple Network Management Protocol (SNMP) for details on SNMP and the CSS.

Table 8-7 Cisco 11500 Series CSS Log Messages

Log Message (sys.log: Subsystem Name, Level, and Message)	Cause and Resolution
NETMAN-5: Generic:SNMP Authentication > Failure from x.x.x.x	A user is trying to use SNMP to poll the CSS, but has entered the wrong community string. Refer to Chapter 12, Configuring Simple Network Management Protocol (SNMP) for details on specifying a community string.
NETMAN-5: Enterprise:Service > Transition:nexthop00001 -> down	The next hop IP address can not be reached by the CSS. When you configure a static route, an internal service is automatically created by the CSS. When the service is up, the static route is included in the routing table. If the service is unavailable, the service is removed from the routing table. Make sure that all of the routes included in the CSS routing table are available. Some routes may have transitioned between states.
NETMAN-5: Generic:LINK UP for 3/1 SYSSOFT-7: NP55_connection.c 512: Connection already open or reserved SYSSOFT-3: NP55 Driver: Connection already open or reserved SYSSOFT-2: VccAddVc:open conn failed w/ stat = -1; iVc 320; eVc 290 FLOWMGR-7: FM_GetIpv4Vc: Warning VCC_FP_IPV4_DC failed	The flow manager tried to reallocate a Virtual Circuit that was already established. These messages occur when the port is coming up. They do not represent a problem. The messages are most noticeable at the end of boot time if you connected through the console.
NETMAN-7: clm_ProcessStdAction:ERROR->Action<clms_dir>not found NETMAN-7: CLM:ERROR from clm_DispatchActionRoutine()	An invalid CLI command was entered. In this example, a user entered the **dir** command in debug mode and specified an invalid directory. For example: (debug)# **dir d:**

Table 8-7 Cisco 11500 Series CSS Log Messages

Log Message (sys.log: Subsystem Name, Level, and Message)	Cause and Resolution

`NETMAN-7: SNMP:UNKNOWN RSP (493512`
`NETMAN-7: SNMP:(493512) Index = 1 <NO_SUCH_NAME>`
	A valid SNMP agent (community string matched) is attempting to set an invalid object and the CSS does not recognize the object.

`NETMAN-7: TSM:tsm_SendToCLA:ERROR->Write`	This security manager message is associated with line data moving through the stack after a line has been disconnected (Telnet application disconnect). This message is at DEBUG level for developer information purposes only.
`NETMAN-7: ASUPPORT:as_SyncTask:ERROR->No registered reciever for MT:5/1.0 W`	This is an informational message. No further action is required.

Table 8-7 Cisco 11500 Series CSS Log Messages

Log Message (sys.log: Subsystem Name, Level, and Message)	Cause and Resolution
portmapper Subsystem	
`PORTMAPPER-5: PortUnmap no Port mapping found.`	A source group is running out of portmappers. Use the **portmap** command to increase the portmappers on the source group so that the message does not appear, and users or services do not see a performance or network address translation problem.
publish Subsystem	
`PUBLISH-1: Unable to allocate tree memory <4150000>`	The CSS is looking for a segment of memory that is approximately 4150000 bytes and it cannot locate an available block of memory. In some cases, this message may be caused by a replication misconfiguration. Review the configuration and verify that it reflects what was intended. Verify that files are being replicated properly. If this message is seen with significantly smaller memory requests, the system memory may not sufficient in size to meet the requirements of the configuration. To isolate the issue, monitor the available memory under non-replication conditions to determine a baseline and then repeat this process while replicating to isolate this issue. Use the **show system-resources** command to view information about the installed and free memory in the CSS. To make additional memory available, reboot the CSS.

Table 8-7 Cisco 11500 Series CSS Log Messages

Log Message (sys.log: Subsystem Name, Level, and Message)	Cause and Resolution
radius Subsystem	

`RADIUS-7: Auth Primary`
`RADIUS-7: The id is 63`
`RADIUS-7: Return Auth Primary`
`RADIUS-7: RADIUS no memory available`
	The RADIUS server does not have the correct attributes set up for the CSS. Refer to Chapter 11, Configuring CSS Remote Access Methods for background information on setting up an ACS radius server.

`RADIUS-4: RADIUS Authentication failed with reason code 2`	In this message, the different codes include:
	`#define PW_ACCESS_REQUEST 1`
	`#define PW_ACCESS_ACCEPT 2`
	`#define PW_ACCESS_REJECT 3`
	`#define PW_ACCOUNTING_REQUEST 4`
	`#define PW_ACCOUNTING_RESPONSE 5`
	`#define PW_ACCOUNTING_STATUS 6`
	`#define PW_ACCESS_CHALLENGE 11`
	In the example above, code 2 indicates that the CSS received the Accept response from the RADIUS server but may have rejected the Accept response, perhaps due to an invalid username or password.
	This log message only appears when logging debug messages (debug-7) for the radius subsystem.

Table 8-7 Cisco 11500 Series CSS Log Messages

Log Message (sys.log: Subsystem Name, Level, and Message)	Cause and Resolution
sntp Subsystem	
`SNTP-6: Sntp Server has incorrect mode 29`	This message indicates potential issues with the SNTP server. Ensure that the SNTP server is transmitting time updates as "server" to the CSS. SNTP server updates are the only SNTP server updates supported by the CSS.
syssoft Subsystem	
`SYSSOFT-2: VccAddVc:open conn failed w/ stat = -1; iVc 320; eVc 290`	This message occurs when a port transitions from an up to a down state. Check autonegotiation, for a defective cable, or for malfunctioning hardware.

`SYSSOFT-3: ONDM: Could not open file <wsscm.sys>`
`SYSSOFT-3: ONDM: Could not download Sub-module 8/1.`
`SYSSOFT-3: ONDM: Could not open file <wssfm.sys>`
`SYSSOFT-3: ONDM: Could not download Sub-module 6/2.`
`SYSSOFT-3: ONDM: Could not download Sub-module 6/1.`
`SYSSOFT-3: ONDM: Could not download Sub-module 5/2`
`SYSSOFT-3: ONDM: Could not download Sub-module 5/1.`
`SYSSOFT-3: ONDM:No Sfm proxy for Slot 2.`
`SYSSOFT-3: ONDM:No Sfm proxy for Slot 1.`
	The CSS could not find the image file to load on the disk. There is something wrong with the disk or the file was deleted from the directory.
	Contact Cisco Systems TAC.

`SYSSOFT-4: SYS:SysImmBind:Bind Collision TSM:5/1.1 W`	This is an informational message. No further action is required.

Table 8-7 Cisco 11500 Series CSS Log Messages

Log Message (sys.log: Subsystem Name, Level, and Message)	Cause and Resolution
`SYSSOFT-4: Invalid Target(0x03087a01) for Chassis Type, Message being dropped.`	The CSS attempts to request the sending of a message to a slot and subslot that does not exist in the chassis. The log message indicates the incorrect address in hexadecimal and the message has been dropped. This message most likely occurs when you issue a command to slot 4 and subslot 1 on a CSS 11503. No corrective action is required.
`SYSSOFT-4: Event not deliverable, msgq id =0x8cc48980, event id = 29, event name = BridgeMacAddrEvent`	The CSS was unable to deliver a certain process because a queue was full. Every message signifies that a event has been dropped because the queue full condition. This message appears when the fastpath (network processor) performs a source MAC address lookup and cannot find an entry. The fastpath then sends a MAC address learn message to the SCM. If the SCM recieves too many messages before it has time to process them, the messages fill the queue.

Table 8-7 Cisco 11500 Series CSS Log Messages

Log Message (sys.log: Subsystem Name, Level, and Message)	Cause and Resolution
`SYSSOFT-4: Event not deliverable, msgq id = 0x865c2110, event id = 4, event name = Ipv4RouteChangeEvent`	The CSS was unable to deliver a certain process because a queue was full. Every message signifies that a event has been dropped because the queue full condition. This message happens whenever a route change is detected. The RIP process, the OSPF process, the caretaker processes (one for each SFM, which try to keep the SFM and SCM route tables in sync), the static route process, and the ARP process register for this event. Look for any routes transitioning state, locally attached stations or servers going up and down, or a large number of ARP requests being performed.
`SYSSOFT-7: MPOOL:mpoolAutoAlloc:WARN->Overrun on MPOOL 3 321`	This message typically appears at boot up as an informational message to let you know that additional CSS memory is being allocated. No further action is required.

Table 8-7 Cisco 11500 Series CSS Log Messages

Log Message (sys.log: Subsystem Name, Level, and Message)	Cause and Resolution
vlanmgr Subsystem	
`VLANMGR-4: DeleteMacAddr() called with VlanID = 0 for MacAddr 0- 0- 0- 0- 0- 0`	The VLAN Manager is being asked to delete a MAC address entry from the forwarding table with a VLAN ID and MAC address of all zeros.
vpm Subsystem	
`VPM removed Vc 8000b00 based on failure of port 3401f00.<010>`	The CSS is reclaiming the resources used by a specific Ethernet port because the port is unavailable. The CSS reclaims resources when a port is unresponsive to an internal check, or when a circuit is unavailable. No addressing information is available for that Ethernet port. Use the **show interface** command to display information for the Ethernet ports and detemine which port is the problem.
wcc Subsystem	
`WCC-7: Route Change for IP Address ( x.x.x.x)`	This is an informational message, stating that an ARP came in on a different port.

Where to Go Next

Chapter 9, Configuring Flow and Port Mapping Parameters provides information on how to configure TCP and UDP flow parameters for the CSS, such as configuring connections for TCP or UDP ports, configuring flow resource reclamation, and configuring CSS port mapping

Configuring Flow and Port Mapping Parameters

This chapter describes how to configure flow and port mapping parameters for the CSS. Information in this chapter applies to all CSS models, except where noted.

This chapter contains the following major sections:

- Overview of CSS Flow
- Configuring Flow Parameters
- Configuring Flow Inactivity Timeouts on Content Rules and Source Groups
- Configuring Flow Processing for Fragmented UDP IP Packets
- Configuring CSS Port Mapping

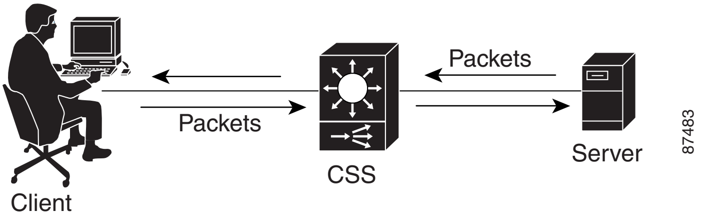
Overview of CSS Flow

A flow is the transfer of a sequence of related packets over a TCP or UDP connection between a source (client) and a destination (server) through the CSS. All packets in an ingress flow (traffic entering the CSS) share a common 5-tuple consisting of:

- Source address
- Destination address
- Protocol
- Source port
- Destination port

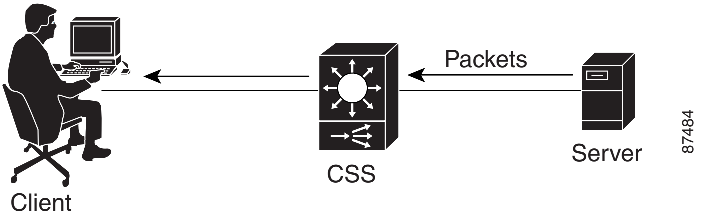
TCP flows are bidirectional (Figure 9-1). Packets move from the client to the server and from the server to the client through the CSS. Strictly speaking, a TCP connection consists of two flows, one in each direction. A TCP flow begins with a SYN and ends with an ACK to a FIN/ACK, or an RST.

Figure 9-1 Example of a TCP Flow



UDP flows (Figure 9-2) are typically unidirectional (for example, streaming audio transmitted by a server to a client). A UDP flow has no definitive beginning or end and is considered completed only after a period of time has elapsed during which the destination device receives no packets that share the same addresses, protocol, and ports that defined the original flow.

Figure 9-2 Example of a UDP Flow



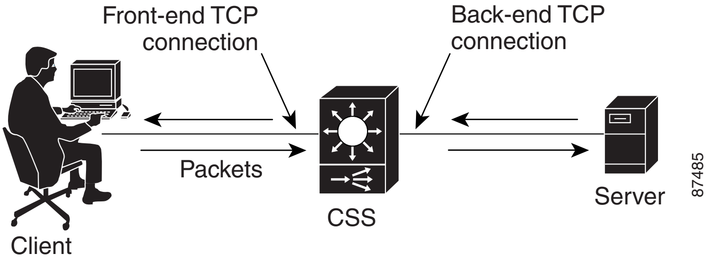
A CSS uses data structures called flow control blocks (FCBs) to set up and keep track of ingress flows. FCBs contain all the information the CSS needs to process and manage flows. The creation of an FCB from flow information is called *flow mapping*. The flow manager in each module session processor is responsible for FCB creation and flow mapping.

Each unidirectional flow uses one FCB. Therefore, a TCP flow uses two FCBs and a UDP flow typically uses one FCB. Front-end SSL, which runs over TCP, requires four FCBs and back-end SSL adds two more FCBs for a total of six FCBs per full-duplex SSL connection. For more information about SLL, refer to the *Cisco Content Services Switch Advanced Configuration Guide*.

Each client-CSS-server connection consists of two parts (Figure 9-3):

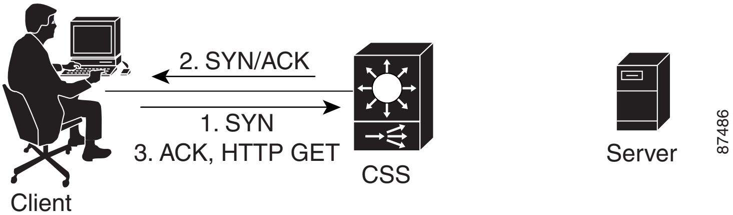
- Front-end - Connection between a client and the CSS
- Back-end - Connection between the CSS and a server

Figure 9-3 Example of a TCP Flow with Front-End and Back-End Connections



A Layer 5 flow begins with a client request for content. After the D-proxy resolves the DNS request (for example, a client types a URL in a Web browser) and points the client to the CSS virtual IP address (VIP), the CSS establishes the front-end TCP connection with the client using the TCP 3-way handshake (Figure 9-4).

Figure 9-4 Setting Up the Front-End TCP Connection - Delayed Binding



When it establishes a Layer 5 flow, a CSS “spoofs” the back-end TCP connection by acting as a proxy for the destination device (server) for the client SYN. In other words, the CSS responds to the client SYN with a SYN/ACK before the CSS sets up the back-end TCP connection with the server.

This process is referred to as *delayed binding*. Delayed binding causes the client to respond with an ACK and an HTTP GET request. This process allows the CSS to gather the information it needs to select the best service (a server port where content resides or an application running on a server such as FTP) for the content request.

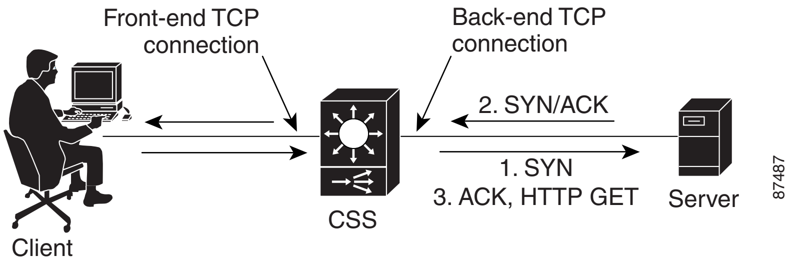
The CSS examines the HTTP header and URL in the HTTP request method (for example, GET, HEAD, or POST). Based on the information in the HTTP header, the URL, and the content rules configured on the CSS, the CSS selects the best site and the best service to satisfy the request. A CSS bases service selection (server load balancing) on factors such as:

- Content rule match
- Service availability
- Service load
- Cookies
- Source IP address

For more information about CSS server load balancing (SLB), refer to the *Cisco Content Services Switch Basic Configuration Guide*.

After the CSS selects the best service to provide the requested content to the client, the CSS establishes the back-end connection with the service using the TCP 3-way handshake and splices the front-end and back-end connections together. The CSS forwards the content request from the client to the service (Figure 9-5). The service responds to the client through the CSS. For the remaining life of the flow, the CSS switches the packets between the client and the service, and performs network address translation (NAT) and other packet transformations as required.

Figure 9-5 Setting Up the Back-End TCP Connection - Delayed Binding



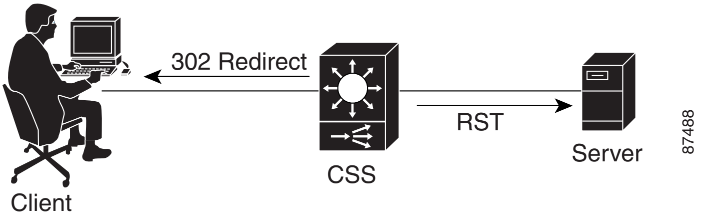
For subsequent content requests from the same client over the same TCP connection (HTTP 1.1 and higher), the CSS attempts to maintain the back-end connection with the service that provided the content for the first HTTP request by default. This condition is called *persistence*.

During the life of a persistent connection, a CSS must determine if it needs to move a client connection to a new service based on content rules, load balancing, and service availability. In some situations, moving the client connection is not necessary; in other situations, it is mandatory.

You can configure the CSS to perform one of the following functions when it becomes necessary to move a client to a new service:

- HTTP redirection - Using the **persistence reset redirect** command, a CSS closes the back-end connection by sending a RST to the service (Figure 9-6). The CSS sends a 302 redirect to the client's browser to tell the browser to reconnect using the same DNS name, but this time the HTTP request matches on a different content rule. The CSS then establishes a new flow between the client and the best service.

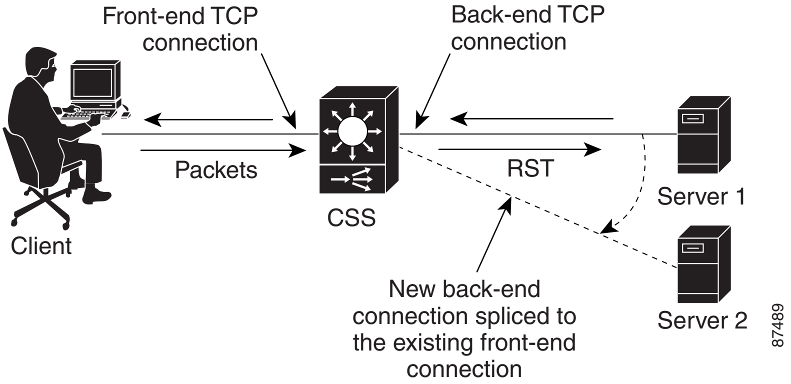
Figure 9-6 Example of HTTP Redirection



- Service remapping - Using the **persistence reset remap** command, a CSS closes only the back-end connection by sending a RST to the service (server 1 in Figure 9-7), then establishes a new back-end connection with service server 2 and splices the back-end and front-end connections together. The CSS forwards the content request from the client to server 2. Packets now flow between the client and server 2.

For more information about persistence, HTTP redirection, and service remapping, refer to the *Cisco Content Services Switch Basic Configuration Guide*.

Figure 9-7 Example of Remapping the Back-end Connection



Periodically, the CSS flow manager tears down old, idle flows and reclaims the system resources (FCBs). This process is called *flow resource reclamation*. It is also referred to as *flow cleanup* or *garbage collection*. Flow resource reclamation involves removing FCBs from the TCP and UDP lists. For optimal performance, the CSS reuses FCBs that are no longer needed for flows.

Normally, flow cleanup occurs at a rate that is directly related to the total number of flows that are currently active on a CSS. A CSS always cleans up UDP flows. For TCP flows, a CSS reclaims resources when the number of used FCBs reaches a certain percentage of the total FCBs. A CSS also cleans up long-lived TCP flows that have received a FIN or a RST, or whose timeout values have been met. You can configure various commands to change the default flow-cleanup behavior of the CSS.

In some instances it may not be desirable for the CSS to clean up idle TCP flows. For example, during a connection to a database server that must permanently remain active even when no data passes through the connection. If you observe the CSS dropping long-lived idle connections that need to be maintained you can configure the following TCP flow commands:

- **flow permanent** command - Creates permanent TCP or UDP ports that are not reclaimed. See the "Configuring Permanent Connections for TCP or UDP Ports" section for details.
- **flow-timeout-multiplier** command - Configures flow inactivity timeout values for TCP and UDP flows on a per content rule and per source group basis. See the "Configuring Flow Timeouts" section for details.

The remainder of this chapter describes the commands you can use to control how the CSS handles and cleans up TCP and UDP flows.

Configuring Flow Parameters

Use the **flow** command to configure flow parameters for the CSS. The options for this global configuration mode command are as follows:

- **flow permanent** - Creates permanent TCP or UDP ports that are not reclaimed
- **flow port-resets** - Resets Fast Ethernet and Gigabit Ethernet ports automatically when the CSS detects that they are not responding
- **flow reserve-clean** - Reclaims interval flows with port numbers less than or equal to 23
- **flow tcp-mss** - Configures the TCP maximum segment size that the CSS expects to receive from the transmitting device
- **flow statistics** - Displays statistics on currently allocated flows

Note Flow parameter setup by the CSS is restricted on the following TCP or UDP ports: 67 (BOOTP server), 68 (BOOTP client), 137 (NETBIOS name service), 138 (NETBIOS datagram service), 161 (SNMP), 162 (SNMP traps), 520 (RIP), and 8089 (restricted UDP only).

This section includes the following topics:

- Configuring Permanent Connections for TCP or UDP Ports
- Configuring TCP Maximum Segment Size
- Reclaiming Reserved Telnet and FTP Control Ports
- Showing Flow Statistics

Configuring Permanent Connections for TCP or UDP Ports

The CSS allows you to configure a maximum of 10 TCP or UDP ports that have permanent connections and will not be reclaimed by the CSS when the flows are inactive. Use the **flow permanent port1** *portnumber* (through **flow permanent port 10** *portnumber*) commands to configure a TCP or UDP port as a permanent connection. Enter a port number from 0 to 65535. The default is 0.

A CSS may reclaim flows that have not received an ACK or content request after approximately 15 seconds. To prevent the CSS from reclaiming flows to a specific source or destination port, specify one of the **flow permanent port** commands and identify the TCP or UDP port number you do not want reclaimed.

For example, to configure port 80 as a permanent connection, enter:

```
(config) flow permanent port1 80
```

To reset the port number for port1 to 0, enter:

```
(config) no flow permanent port1
```

We recommend that when you configure a **flow permanent port** command you also enable the **cmd-sched** command to periodically remove the permanent port and allow for cleanup. For details on using the **cmd-sched** command to configure the scheduled execution of any CLI command, refer to the *Cisco Content Services Switch Advanced Configuration Guide*.

Configuring TCP Maximum Segment Size

Use the **flow tcp-mss** command to configure the TCP maximum segment size (MSS) that the CSS expects to receive from the transmitting device. The MSS is the largest amount of TCP data that can be transmitted in one segment. The need for a smaller MSS between devices may be necessary in rare instances due to network restrictions between devices. The **flow tcp-mss** command changes the MSS value in the TCP header OPTIONS field of a SYN segment, to reduce the MSS from the default value of 1460 bytes.

The **flow tcp-mss** command applies only when the client is accessing a Layer 5 content rule. The CSS does not negotiate TCP maximum segment size for Layer 3 or Layer 4 content rules.

Enter a maximum segment size (in bytes) from 1 to 1460. The default is 1460 bytes. Use the **no** form of the command to reset the TCP maximum segment size back to the default value of 1460 bytes.

⚠ **Caution** Do not define a smaller than necessary TCP maximum segment size with the **flow tcp-mss** command. Smaller payloads may be less efficient due to increased overhead.

To configure a TCP maximum segment size of 1400 bytes, enter:

```
(config)# flow tcp-mss 1400
```

To reset the TCP maximum segment size to the default value of 1460 bytes, enter:

```
(config)# no flow tcp-mss
```

Reclaiming Reserved Telnet and FTP Control Ports

Use the **flow reserve-clean** command to define how often the CSS scans flows from reserved Telnet and FTP control ports to reclaim them. Control ports have port numbers less than or equal to 23. When the CSS determines that one of these ports has a flow with asymmetrical routing, it reclaims the port.

Enter the **flow reserve-clean** time, in seconds, as the interval the CSS uses to scan flows. Enter an integer from 0 to 100. The default is 10. To disable the port reclaiming process, enter a flow reserve-clean value of 0.

For example, to specify an interval of 36 seconds:

```
(config)# flow reserve-clean 36
```

To disable flow cleanup on Telnet and FTP control ports, enter:

```
(config)# no flow reserve-clean
```

Showing Flow Statistics

Use the **flow statistics** command to display statistics on currently allocated flows.

For example:

```
(config)# flow statistics

Flow Manager Statistics:

                         Current   High     Avg
UDP Flows per second     0         0        0
TCP Flows per second     0         4        0
Total Flows per second   0         4        0
Hits per second          0         0        0

--------------------------------------------------------------
Port      Active      Total         TCP      UDP
--------------------------------------------------------------
1         13          43339169      13       0
2         16          43337519      16       0
5         18          3167362       18       0
6         9           33483528      9        0
```

Configuring Flow Inactivity Timeouts on Content Rules and Source Groups

Use this feature with a CSS to configure flow inactivity timeout values for TCP and UDP flows on a per content rule and per source group basis. This timeout value is *not* the frequency with which a CSS reclaims flow resources, but is the time period that must elapse for an idle flow before the CSS marks the flow for cleanup.

Timeout Value Precedence

The CSS uses the following guidelines in the order presented when reclaiming flow resources:

1. If a flow matches on a content rule, the CSS checks for a user-configured timeout value and uses that timeout value if one exists.
2. If the flow matches on a source group, the CSS checks for a user-configured timeout value and uses that timeout value if one exists.
3. If you have configured a permanent port using the **flow permanent port** command (see the "Configuring Permanent Connections for TCP or UDP Ports" section), the CSS sets the flow timeout value to 0, which means that the flow should never time out.
4. If none of the above conditions are met, then the CSS uses the default timeout value for the protocol type. For more information, see the "Displaying Flow Timeout Statistics" section.

Configuring Flow Timeouts

Use the **flow-timeout-multiplier** command to specify the number of seconds for which an idle flow can exist before the CSS tears it down. Specify this command in owner-content or group configuration mode.

The syntax for this command is:

flow-timeout-multiplier *number*

Enter an integer for the *number* variable from 0 to 65533. The CSS multiplies the value you specify by 16 to calculate the flow timeout in seconds. The default value depends on the TCP or UDP port number (see the "Displaying Flow Timeout Statistics" section). This default value applies only to flows that you create under a content rule or source group.

A value of zero (no timeout) instructs the CSS to never tear down the flow, resulting in a permanent flow and lost resources. Specifying a value of zero is equivalent to entering the **flow permanent port** command (see the "Configuring Permanent Connections for TCP or UDP Ports" section).

Note We do not recommend that you set the **flow-timeout multiplier** command to 0 for UDP flows on Layer 3 and Layer 4 content rules. If the value is set to 0, the CSS does not clean up the resources for the UDP flows.

These two examples show flow timeout periods of 80 seconds:

```
(config-owner-content[cisco-rule1])# flow-timeout-multiplier 5
(config-group[group1])# flow-timeout-multiplier 5
```

To disable the configured **flow-timeout-multiplier** value and restore the default timeout for the port type, enter:

```
(config-owner-content[cisco-rule1])# no flow-timeout
(config-group[group1])# no flow-timeout
```

Displaying Flow Timeout Statistics

Use the **show flow-timeout default** command to display the default timeout values for TCP and UDP ports and applications. The default values are not user configurable.

Table 9-1 shows the fields in the **show flow-timeout default** command output.

Table 9-1 Field Descriptions for show flow-timeout default Command

Field	Description
TCP/IP Port	Default TCP or UDP port numbers.
Application	Names of the default TCP or UDP applications.
Inactivity Timeout Seconds	Default flow inactivity timeouts, in seconds, for the TCP or UDP port. If a flow is idle for the amount of time specified in the timeout value, the CSS tears down the flow and reclaims the flow resources.

Use the **show flow-timeout configured** command to display the configured flow timeout values. The command output includes the content rule or source group for which you configured the flow timeout value.

Table 9-2 describes the fields in the **show flow-timeout configured** command output.

Table 9-2 Field Descriptions for the show flow-timeout configured Command

Field	Description
Port	TCP or UDP port number.
Content Rule	Name of the content rule for which the flow timeout is configured.
Source Group	Name of the source group for which the flow timeout is configured.
Timeout	Configured inactivity timeout in 16-second increments for the TCP or UDP port. When this time period elapses for an idle flow, the CSS tears down the connection and reclaims the FCBs.

Displaying Content Rule and Source Group Information

If you configure a flow timeout value in a content rule or a source group, the **show rule** or **show group** command output includes an additional field called Flow Timeout Multiplier. This field contains the configured timeout value assigned to flows that match on the rule or group.

For details on the **show rule** command and the **show group** command, refer to the *Cisco Content Services Switch Basic Configuration Guide*.

Configuring Flow Processing for Fragmented UDP IP Packets

By default, a CSS does not process fragmented UDP IP packets (IP fragments) in the flow path, but simply routes them according to standard IP routing practices. As a result, IP fragments do not match on configuration items such as content rules and source groups.

When you enable flow processing for IP fragments, a CSS processes the IP fragments in the flow path using the IP address and UDP port information in the IP and UDP headers. The CSS then forwards and NATs the individual fragments of a packet based on the configured content rules and source groups matched by the fragments.

Note Whenever possible, avoid applications or network configurations that create IP fragments. This feature provides support for those edge conditions where IP fragmentation is unavoidable.

Use this feature to support:

- Microsoft Media Server using Microsoft Media Server UDP (MMSU) protocol
- Network configurations where UDP IP packets must be fragmented to traverse the network

This section describes how to configure flow processing for fragmented IP UDP packets. It includes the following topics:

- What Is IP Packet Fragmentation?
- Enabling Flow Processing for Fragmented UDP IP Packets
- Configuring the Maximum Assembled Size
- Configuring the Minimum Fragment Size
- Resetting IP Fragment Statistics
- Displaying IP Fragment Statistics

What Is IP Packet Fragmentation?

An IP fragment is a part of a larger complete IP packet. IP packets require fragmentation when the next-hop network's maximum transmit unit (MTU) is less than the incoming packet size. The transmitting device divides the packet into smaller pieces that the network medium can accommodate and copies the packet IP header into each fragment. Packets can be fragmented by the source host, intermediate routers, and other network devices, such as the CSS.

IP packet fragmentation is generally considered an undesirable condition because fragmentation and subsequent reassembly of packets cause additional CPU and network overhead. However, despite the best efforts of network designers, some fragmentation is inevitable because of the different network media with varying MTUs that support the IP protocol.

For more information about IP packet fragmentation and reassembly, refer to RFC 791 and RFC 815.

Enabling Flow Processing for Fragmented UDP IP Packets

Note This feature supports only IP UDP fragment. Using this feature, the CSS does not process IP TCP fragments and IP fragments from other protocols, but it will continue to route such fragments.

Note This feature performs content rule-based forwarding using the Layer 3 (IP address) and Layer 4 (UDP port) information in the IP and UDP headers. Layer 5 forwarding decisions for IP UDP fragments, based on the packet payload (data), are not supported.

To allow a CSS to flow-process IP UDP fragments, use the **ip-fragment-enabled** command in global configuration mode. By default, this feature is disabled.

To reset the default behavior of the CSS to forward IP fragments, use the **no** form of the command.

For example, enter:

```
(config)# no ip-fragment-enabled
```

Configuring the Maximum Assembled Size

The maximum assembled size is the total length of an IP packet if all the IP fragments were assembled into the original packet. Assembled IP packets should be no larger than 64 KB. As the CSS receives the IP fragments, it checks the fragments against the maximum assembled size value. If a fragment IP offset plus the IP payload (data) length is greater than the configured maximum assembled size, the CSS increments the Max Assembled Size error field in the **show ip-fragment-stats** command output and discards the packet. See the "Displaying IP Fragment Statistics" section.

To specify the maximum assembled size, use the **ip-fragment max-assembled-size** command. The syntax of this global configuration mode command is:

ip-fragment max-assembled-size *number*

The *number* variable specifies the maximum size of an assembled packet in bytes. Enter an integer from 2048 to 65535. The default is 5120 bytes.

For example, enter:

```
(config)# ip-fragment max-assembled-size 4096
```

To restore the default maximum IP fragment assembled size to 5120 bytes, use the **no** form of the command.

For example, enter:

```
(config)# no ip-fragment max-assembled-size
```

Configuring the Minimum Fragment Size

The minimum fragment size is the smallest IP payload in an IP fragment that a CSS accepts. As the CSS receives the IP fragments, it checks the fragments against the minimum fragment size value. If a fragment IP payload length is less than the configured minimum fragment size, the CSS increments the Less Than Min Size error field in the **show ip-fragment-stats** command output and discards the packet. See the "Displaying IP Fragment Statistics" section.

To specify the smallest IP fragment payload based on your applications, use the **ip-fragment min-fragment-size** command. This command also provides protection against fragment attacks, which can consist of a chain of valid-looking, but very small, fragments.

The syntax of this global configuration mode command is:

ip-fragment min-fragment-size *number*

The *number* variable specifies the size of the smallest IP fragment payload that the CSS supports in bytes. Enter an integer from 64 to 1024. The default is 1024 bytes.

For example, enter:

```
(config)# ip-fragment min-fragment-size 256
```

Note Requiring that the minimum fragment size be at least 64 bytes guarantees that the IP and UDP header information is present in the first fragment.

To restore the default minimum IP fragment payload size to 1024 bytes, use the **no** form of the command.

For example, enter:

```
(config)# no ip-fragment min-fragment-size
```

Resetting IP Fragment Statistics

To reset the IP fragment statistics, use the **zero ip-fragment-stats** command in any mode. This command resets the values of the statistics in the IP Fragment Statistics and IP Fragment Errors sections of the **show ip-fragment-stats** command output to zero.

For more information about the **show ip-fragment-stats** command, see the "Displaying IP Fragment Statistics" section.

Displaying IP Fragment Statistics

To display the status, statistics, and error counts associated with IP fragment processing, use the **show ip-fragment-stats** command in any mode.

Table 9-3 describes the fields for the **show ip-fragment-stats** command output.

Table 9-3 Field Descriptions for the show ip-fragment-stats Command

Field	Description
IP Fragment Status	
State	Configured state of the IP fragment feature.
Min Fragment Size	Configured minimum fragment IP payload size.
Max Assembled Size	Configured maximum assembled IP packet size.
IP Fragment Statistics	
Packets Tracked	Number of fragmented IP packets tracked. This field contains the number of actual packets tracked, not the number of fragments.
Fragments Buffered	Number of buffered IP fragments from all tracked packets.
Packets Completed	Number of successfully processed IP packets that were fragmented.
Longest Frag Chain	Longest IP fragment chain that constituted any one fragmented IP packet. An IP fragment chain is the number of fragments that make up the original packet.
Largest Asm Packet	Largest IP length of an IP fragmented packet that the CSS received.
Smallest Fragment	Smallest fragment IP payload length that the CSS received. This field does not include the last fragment in any IP fragment because its payload can be any size.

Table 9-3 Field Descriptions for the show ip-fragment-stats Command (continued)

Field	Description
IP Fragment Errors	
No Tracking Entry	While receiving a fragment of a new packet, the CSS could not obtain a fragment tracking entry. This error can occur if the CSS memory is low or used completely.
Could Not Buffer	CSS received a fragment, but could not buffer it because the CSS was low on buffers.
Duplicate Fragment	CSS detected a duplicate offset or last fragment.
Validating Fragments	After the CSS received all the IP fragments, it attempted to validate the fragments, but found overlapping offsets, short offsets, or other possible denial of service (DoS) fragment attack conditions.
Inserting Fragment	While the CSS was inserting fragments into the fragment chain on the tracking entry, it encountered duplicate fragments, fragments of less than the configured minimum fragment size, or a total assembled size greater than the configured maximum assembled size.
Less Than Min Size	CSS received an IP fragment (not the last fragment) with an IP payload that was less than the configured minimum fragment size.
Max Assembled Size	After the CSS received a fragment, the calculated total length of the assembled IP packet was greater than the configured maximum assembled size.
Collection Timeout	While the CSS was waiting to receive IP fragments, too much time elapsed.
Flow Timeout	After the CSS received all fragments of an IP packet and a fragment was sent to flow processing, the entry timed out before the fragment returned.

Table 9-3 Field Descriptions for the show ip-fragment-stats Command (continued)

Field	Description
IPv4 Header	The CSS received a fragment with an invalid IPv4 header length compared with the total IP fragment size.
RxQueue Full	The CSS flow-processing receive queue for IP fragments was full. The CSS discarded the IP fragments.

Configuring CSS Port Mapping

This section describes how to globally control the range of port numbers a CSS uses to translate (port map) TCP and UDP source port numbers specified in packets sent to the CSS from clients. The CSS assigns unique port numbers within a configurable range for the source port numbers specified in the packets, then sends the packets to the appropriate server port. When a server initiates a return flow, the packets flow through the CSS. The CSS matches the translated port number with the client that initiated the request and sends the server packets to the appropriate client.

Each CSS maintains a database of used and available port-map numbers. When a CSS needs to port map a source port, it uses the next unused port number in its database.

For information about port mapping with source groups, refer to the *Cisco Content Services Switch Basic Configuration Guide.*

This section includes the following topics:

- Configuring Global Port Mapping
- Displaying Global Port Mapping Statistics
- Configuring No-flow Port Mapping
- Displaying No-Flow Port Mapping Statistics

Configuring Global Port Mapping

The global port mapper in a CSS is called the mega port mapper. The mega port mapper database comprises 16 banks of port-map numbers (megamap banks) in each session processor (SP) with unique ranges. A CSS uses a source port hash algorithm to select a megamap bank.

Use the **global-portmap** command to control the global source-port translation (port mapping) for TCP flows on a CSS. This command is always enabled.

You can use this command to specify the source-port mapping range on:

- A CSS when you configure a service that uses a nondefault destination port number. A CSS changes a TCP destination port number configured on a service in a content rule when a request hits the content rule and the CSS sends a packet to the selected server. The CSS uses the **global-portmap** command parameters to translate the corresponding client source port number to distinguish it from other clients requesting the same service.
- Redundant Cisco 11500 series CSS peers in an Adaptive Session Redundancy (ASR) configuration. Refer to the *Cisco Content Services Switch Advanced Configuration Guide* for information about ASR.
- A CSS with back-end server remapping enabled (refer to the *Cisco Content Services Switch Basic Configuration Guide*.

When you configure a source group, the **portmap** command parameter values take precedence over the **global-portmap** command parameter values. The **portmap disable** command has no effect on TCP flows.

The syntax for this global configuration mode command is:

global-portmap base-port *number1* **range** *number2*

The options and variables for this command are as follows:

- **base-port** *number1* - The starting port number for global port mapping on a CSS. Enter an integer from 2016 to 63456. The default is 2016.
- **range** *number2* - The number of ports in the port-map range. Enter an integer from 2048 to 63488. The default is 63488.

Note If you enter a port-map range that exceeds the number of available ports, you get an error. To determine the number of available ports, subtract the starting port number you specify from 65504.

For example:

```
(config)# global-portmap base-port 3096 range 42308
```

To return the global-portmap command parameters to their default values, enter:

```
(config)# no global-portmap
```

Displaying Global Port Mapping Statistics

Use the **show global-portmap** command to display statistics for global port mapping on a CSS. This command is available in all modes except RMON, URQL, and VLAN configuration modes.

The syntax for this command is:

show global-portmap [**all-banks** [**all-sps**|**slot** *number1*]|*number2* [**all-sps**|**slot** *number1*]]

The options and variables for this command are as follows:

- **all-banks** - Specifies the display of global port-map information for all port-map banks (0 to 15).
- **all-sps** - Specifies the display of global port-map information for all session processors (SPs) in the CSS.
- **slot** *number1* - Specifies the chassis slot where the module resides. For a CSS 11503, enter an integer from 1 to 3. For a CSS 11506, enter an integer from 1 to 6.

  To display the available active slots in the CSS, enter the **show global-portmap all-banks slot ?** command. If you enter an invalid slot number, the CLI displays values for only the first two parameters listed in Table 9-4.
- *number2* - Specifies the global port-map bank number. Enter an integer from 0 to 15.

To display port mapping statistics for all megamap banks (up to 16) on every active SP in the CSS, enter:

```
(config)# show global-portmap all-banks all-sps
```

To display global port mapping statistics for megamap bank 12 in the SP that resides in slot 3, enter:

```
(config)# show global-portmap 12 slot 3
```

Table 9-4 describes the fields in the **show global-portmap** command output.

Table 9-4 Field Descriptions for show global-portmap Command

Field	Description
MegaMap Banks in Use Per SP	The number of global port mapping banks being used in each session processor (SP). There are 16 banks available in each SP. A CSS selects a bank by hashing the source address contained in a packet.
Configured Base Port	The **base-port** (starting port number) specified with the **global-portmap** command or the default of 2016.
Total Configured Ports	The total number of ports specified with the **global-portmap range** command or the default of 63488.
Slot	The number of the slot in the CSS 11503 or CSS 11506 where the specified SP resides.
MegaMap Bank #	The number of the port mapping bank. Possible values are 0 to 15 for a total of 16 banks for each SP.
Number Normal Avail Ports	The number of ports available for use by the network address translation algorithm when the source port number is different from the destination port number in a TCP packet.
Current Mapped Ports	The total number of ports currently in use or mapped.
Last Normal Mapped Port	The most recent port number used by the network algorithm when the source port number is different from the destination port number in a TCP packet.
Equal Port Base Port	The starting port number that the network address translation algorithm uses when the source port number is the same as the destination port number in a TCP packet.

Table 9-4 Field Descriptions for show global-portmap Command (continued)

Field	Description
Number Equal Avail Ports	The number of ports available for use by the network address translation algorithm when the source port number is the same as the destination port number in a TCP packet.
High Water Mark	The largest number of ports mapped or in use at one time since the last CSS reboot.
Last Equal Mapped Port	The last port number used by the network address translation algorithm when the source port number is the same as the destination port number in a TCP packet.
No Portmap Errors	The number of times that a failure occurred because no ports were available (all ports were mapped).

Configuring No-flow Port Mapping

Use the **noflow-portmap** command to control the port translation (port mapping) range of DNS UDP source-port numbers greater than 1023 on a CSS. This command is always enabled. However, before a CSS can use this command, you must enter the **dnsflow disable** command to disable DNS flows on the CSS. Refer to Chapter 6, Configuring CSS Network Protocols for details on the **dnsflow** command.

Note The **portmap** command values configured in a source group take precedence over the **noflow-portmap** command values, unless you configure the **portmap disable** command. Refer to the *Cisco Content Services Switch Basic Configuration Guide* for details on configuring the **portmap** commands in a source group.

The syntax for this global configuration mode command is:

noflow-portmap base-port *number1* **range** *number2*

The options and variables for this command are as follows:

- **base-port** *number1* - The starting port number for no-flow (DNS flows are disabled) port mapping on a CSS. Enter an integer from 2016 to 63456. The default is 2016.

- **range** *number2* - The number of ports in the port-map range. Enter an integer from 2048 to 63488. The default is 63488.

Note If you enter a value for the port-map range that exceeds the number of available ports, you get an error. To determine the number of available ports, subtract the starting port number from 65504.

For example, to specify a port map range, starting with port 4317, enter:

```
(config)# noflow-portmap base-port 4317 range 35421
```

To reset the starting port number and port-map range to their default values, enter:

```
(config)# no noflow-portmap
```

Displaying No-Flow Port Mapping Statistics

Use the **show noflow-portmap** command to display statistics for no-flow port mapping on a CSS. This command is available in all modes except RMON, URQL, and VLAN configuration modes.

The syntax for this command is:

show noflow-portmap [**all-sps**|**slot** *number*]

The options and variables for this command are as follows:

- **all-sps** - Specifies the display of no-flow port-map information for all session processors (SPs) in the CSS.
- **slot** *number* - The chassis slot number where the module resides. For a CSS 11503, enter an integer from 1 to 3. For a CSS 11506, enter an integer from 1 to 6.

Note To display the available active slots in the CSS, enter the **show noflow-portmap slot ?** command. If you enter an invalid slot number, the CLI displays values for only the first two parameters listed in Table 9-5.

For example:

```
(config)# show noflow-portmap slot 3
```

Table 9-5 describes the fields in the **show noflow-portmap** command output.

Table 9-5 Field Descriptions for show noflow-portmap Command

Field	Description
Configured Base Port	The starting port number specified by the **noflow-portmap base-port** command or the default of 2016
Total Configured Ports	The total number of ports specified by the **noflow-portmap range** command or the default of 63488
Slot	The number of the slot in the CSS 11503 or CSS 11506 where the specified SP resides
Number Normal Avail Ports	The number of ports available for use by the network address translation algorithm when the source port number is different from the destination port number in a UDP packet
Current Mapped Ports	The total number of ports currently in use or mapped
Last Normal Mapped Port	The most recent port number used by the network address translation algorithm when the source port number is different from the destination port number in a UDP packet
Equal Port Base Port	The starting port number that the network address translation algorithm uses when the source port number is the same as the destination port number in a UDP packet
Number Equal Avail Ports	The number of ports available for use by the network address translation algorithm when the source port number is the same as the destination port number in a UDP packet
High Water Mark	The largest number of ports mapped or in use at one time since the last CSS reboot.
Last Equal Mapped Port	The last port number used by the network address translation algorithm when the source port number was the same as the destination port number in a UDP packet
No Portmap Errors	The number of times that a failure occurred because no ports were available (all ports were mapped)

Where to Go Next

Chapter 10, Configuring User Profiles provides information about how to configure CSS user profiles in the default-profile file.

CHAPTER 10

Configuring User Profiles

This chapter describes how to configure user profiles. Information in this chapter applies to all models of the CSS, except where noted.

This chapter contains the following major sections:

- User Profiles Overview
- Configuring User Terminal Parameters
- Configuring Idle Timeout
- Using Expert Mode
- Changing the CLI Prompt
- Modifying the History Buffer
- Copying and Saving User Profiles
- Configuring a Login Banner

User Profiles Overview

The CSS contains a default-profile file that resides in the scripts directory on the CSS disk (hard disk or Flash disk). This file contains settings that are user-specific; that is, they apply uniquely to each user when that user logs in.

You can customize the following settings for each user:

- CLI prompt
- Expert mode
- History buffer
- Terminal parameters
- Login banner

Though the settings are user-specific, each default setting applies to all users until the user saves the default-profile file to a *username*-profile (where *username* is the current login username). You can continue using the default-profile file to ensure all users logging in to a CSS use the same settings. See the "Copying and Saving User Profiles" section for information on saving the default-profile file.

If you change a user setting and want to save this setting in the scripts directory of the current ADI, use the **copy profile** command. If you do not save this setting, the CSS stores the setting temporarily in a running profile. If you attempt to log out of the CSS without saving profile changes, the CSS prompts you that profile changes have been made and allows you to save or discard the changes.

When you upgrade the ADI, the CSS deletes all user profile from the current ADI directory. If you wish to save user profiles permanently, use the **save_profile** command. This command saves the profiles in both the scripts and archive directories in the current ADI. The archive directory is not overwritten during a software upgrade.

To access the CSS disk, FTP to the CSS. Use the appropriate commands to access the scripts directory and list the contents of the default-profile file. When logged into the CSS, use the **show profile** command to display either the default-profile file or your *username*-profile file.

For example:

```
# show profile
```

```
@prompt CSS11503
@no expert
alias all reboot "@configure;boot;rebo"
alias all shutdown "@configure;boot;shutd"
alias all logon "@configure;logging line \${LINE};exit"
alias all logoff "@configure;no logging line \${LINE};exit"
alias all aca-load "@script play service-load"
alias all dnslookup "@script play dnslookup"
alias super save_config "copy running-config startup-config;archive
startup-config"
alias super setup "script play setup"
alias super upgrade "script play upgrade"
alias super monitor "script play monitor"
alias super save_profile "copy profile user-profile;archive script
admin-profile
"
set CHECK_STARTUP_ERRORS "1" session
```

Configuring User Terminal Parameters

Use the **terminal** command to configure terminal parameters. These parameters control output to the system terminal screen. Terminal parameters are user-specific; that is, they apply uniquely to each CSS user.

Use the **copy profile user-profile** command to add terminal command parameters to your user profile so that the parameters are used each time you log in. Otherwise, you must reenter the commands for the parameters to take effect each time you log in.

This section includes the following topics:

- Configuring Terminal Idle
- Configuring Terminal Length
- Configuring the More Terminal Prompt
- Configuring Terminal Netmask-Format
- Configuring Terminal Timeout

Configuring Terminal Idle

Use the **terminal idle** command to set the length of time a session can be idle before the CSS terminates a console or Telnet session. This command is available in the User and SuperUser modes. Enter an idle time between 0 and 65535 minutes. The default value is 0 (disabled).

To set a terminal idle time, enter:

```
# terminal idle 15
```

To restore the terminal idle time to the default state of disabled, enter:

```
# no terminal idle
```

Configuring Terminal Length

Use the **terminal length** command to set the number of output lines the CLI displays on the terminal screen. This command is available in User and SuperUser modes. Enter the number of lines you want the CLI to display, from 2 to 65535. The default is 25 lines.

To set the line number to 35, enter:

```
# terminal length 35
```

To reset the number of lines to the default of 25 lines, enter:

```
# no terminal length
```

Configuring the More Terminal Prompt

Use the **terminal more** command to display the --More-- prompt at the bottom of the terminal screen. When you enter the question mark (?) character at the command line to get help about a command, the CSS displays 24 lines on the terminal. The --More-- prompt indicates that additional CLI commands are to follow. Press the **Space** bar to continue viewing the next series of commands (or press the **Return** key to display only the next line). This command is available in User and SuperUser modes. The default is enabled.

You can also toggle the terminal more function on and off within a session by using the Esc-M key sequence.

To enable support for the --More-- terminal prompt, enter:

```
# terminal more
```

To disable support for the --More-- terminal prompt, enter:

```
# no terminal more
```

Configuring Terminal Netmask-Format

Use the **terminal netmask-format** command to determine how the CSS displays subnet masks in show screens. This command is available in User and SuperUser modes. The options for this command are as follows:

- **terminal netmask-format bitcount** - Displays masks in classless interdomain routing (CIDR) bitcount (for example, /24).
- **terminal netmask-format decimal** - Displays masks in dotted-decimal format (for example, 255.255.255.0). This is the default format.
- **terminal netmask-format hexadecimal** - Displays masks in hexadecimal format (for example, OXFFFFFFOO).

For example, to display subnet masks in bit-count format, enter:

```
# terminal netmask-format bitcount
```

To restore the default display format (**decimal**), enter:

```
# no terminal netmask format
```

Configuring Terminal Timeout

Use the **terminal timeout** command to set the total amount of time a session can be logged in before the CSS terminates a console or Telnet session. This command is available in User and SuperUser modes. Enter a timeout value between 0 and 65535 minutes. The default value is 0 (disabled).

For example, to set the terminal timeout value to 30 minutes, enter:

```
# terminal timeout 30
```

To restore the terminal timeout value to the default state of disabled, enter:

```
# no terminal timeout
```

Configuring Idle Timeout

Use the **idle timeout** command to globally set the total amount of time all sessions can be active before the CSS terminates a console or Telnet session. Enter a timeout value between 0 and 65535 minutes. The default value is enabled for 5 minutes.

To override the idle timeout value for a specific session, configure the **terminal timeout command**. Terminal commands are user-specific; that is, they apply uniquely to each CSS user.

We recommend that you configure the idle timeout for at least 30 minutes. Setting this value to 30 minutes:

- Cleans up idle Telnet sessions
- Helps prevent busy conditions due to a high number of active Telnet sessions

To set an idle timeout value, enter:

```
(config)# idle timeout 15
```

To restore the terminal timeout value to the default of enabled for 5 minutes, enter:

```
(config)# no idle timeout
```

Using Expert Mode

Expert mode allows you to turn the CSS confirmation capability on or off. Expert mode is available in SuperUser mode and is off by default. When expert mode is off, the CSS prompts you for confirmation when you:

- Execute commands that could delete or change operating parameters
- Exit a terminal session (console or Telnet) without copying the running configuration to the startup configuration
- Create services, owners, and content rules

Caution Turning expert mode on *disables* the CSS from prompting you for confirmation when you make changes. When you exit from the CSS, all configuration changes are automatically saved to the profile and to the running-config file. You are not prompted for confirmation to save the changes.

To enable expert mode, enter:

```
# expert
```

To allow the CSS to prompt you for confirmation before executing configuration commands, enter:

```
# no expert
```

For example, when you enter the command to create an owner and expert mode is off, the CSS prompts you to verify the command, enter:

```
(config)# owner arrowpoint.com
Create owner <arrowpoint.com>, [y/n]:y
(config-owner[arrowpoint.com])#
```

Changing the CLI Prompt

The CLI default prompt appears as the CSS product model number followed by the number (#) symbol. The CSS adds a **#** to the prompt automatically to indicate SuperUser mode. To change the default prompt, enter the **prompt** command. For example:

```
CSS11506# prompt CSS1-lab
CSS1-lab#
```

You can enter a maximum of 15 characters.

To save the new prompt, add it to your user- or default-profile file. To restore the prompt to the default, use the **no** form of the **prompt** command.

For example:

```
CSS11506# no prompt
```

Modifying the History Buffer

Use the **history** command to modify the CLI history buffer length. The history buffer stores the most recent CLI commands that you enter. Enter the number of lines you want in the history buffer as an integer from 0 to 256. The default is 20. This command is available only in SuperUser mode.

To set the history buffer to 80 lines, enter:

```
# history length 80
```

To disable the history function (setting of 0), enter:

```
# history length 0
```

To restore the history buffer to the default of 20 lines, enter:

```
# no history length
```

Displaying the History Buffer

Use the **show history** command to display the contents of the history buffer. The history buffer is cleared automatically upon reboot.

For example:

```
# show history

   history
   show history
   show ip routes
   show ip summary
   show ip stat
   clock
   clock date
   clock time
   show history
```

Copying and Saving User Profiles

Use the **copy profile** command to copy the running profile from the CSS to the default-profile file, an FTP server, a TFTP server, or your user-profile file. This command is available only in SuperUser mode.

If you exit the CSS without copying changes in the running profile to your *username*-profile or default-profile file, the CSS prompts you that the profile has changed and queries whether you want to save your changes. If you respond with **y**, the CSS copies the running profile to your *username*-profile or the default-profile.

This section includes the following topics:

- Copying the Running Profile to the Default-Profile
- Copying the Running Profile to a User Profile
- Copying the Running Profile to an FTP Server
- Copying the Running Profile to a TFTP Server

Copying the Running Profile to the Default-Profile

Use the **copy profile default-profile** command to copy the running profile to the default profile. This command is available only in SuperUser mode.

For example:

```
# copy profile default-profile
```

Copying the Running Profile to a User Profile

Use the **copy profile user-profile** command to copy the changes made to the running profile to the user profile. This command is available only in SuperUser mode. This command creates a file *username*-profile if one does not exist (where *username* is the current username).

For example:

```
# copy profile user-profile
```

Copying the Running Profile to an FTP Server

Use the **copy profile ftp** command to copy the running profile to an FTP server. This command is available only in SuperUser mode. The syntax is:

copy profile ftp *ftp_record filename*

The variables for this command are as follows:

- *ftp_record* - The name of the FTP record file that contains the server IP address, username, and password. Enter an unquoted text string with no spaces and a maximum of 32 characters.
- *filename* - The name you want to assign to the file on the server. Include the full path to the file. Enter an unquoted text string with no spaces.

For example:

```
# copy profile ftp arrowrecord \records\arrowftprecord
```

Copying the Running Profile to a TFTP Server

Use the **copy profile tftp** command to copy the running profile to a TFTP server. This command is available only in SuperUser mode. The syntax is:

```
copy profile tftp ip_or_host filename
```

The variables for this command are as follows:

- *ip_or_host* - The IP address or host name of the server to receive the file. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1) or in mnemonic host-name format (for example, myhost.mydomain.com).
- *filename* - The name you want to assign to the file on the server. Include the full path to the file. Enter an unquoted text string with no spaces and a maximum of 32 characters.

For example:

```
# copy profile tftp 192.168.3.6 \home\bobo\bobo-profile
```

Configuring a Login Banner

Use the **set BANNER** command to create and display a custom banner that appears after you log in to a CSS. The banner is an ASCII text file that you provide, and it resides in the CSS script directory. Because this feature is part of your user profile, you can have your own custom banner associated with your login name.

To configure a login banner, use the **set BANNER** command as follows.

1. Use any text editor (for example, Notepad or Wordpad) to create a custom banner and save it as a text file in the CSS script directory. Line width has a maximum of 80 characters.
2. Set the environment variable BANNER and point to the *mybanner* file:

   ```
   # set BANNER “mybanner” session
   ```

   The keyword **session** causes the CSS to create the environment variable every time you log in.

3. Enter the following to complete the configuration:

```
# copy profile user-profile
```

   The next time you log in to the CSS, you see your custom banner.

To provide the same banner for all users, replace the command in Step 3 with:

```
# copy profile default-profile
```

Where to Go Next

Chapter 11, Configuring CSS Remote Access Methods provides information on configuring CSS remote access methods, including the Secure Shell Daemon (SSH) protocol, the Remote Authentication Dial-In User Service (RADIUS) protocol, and the Terminal Access Controller Access Control System (TACACS+) protocol.

CHAPTER 11

Configuring CSS Remote Access Methods

This chapter describes how to configure the Secure Shell Daemon (SSH), Remote Authentication Dial-In User Service (RADIUS), and the Terminal Access Controller Access Control System (TACACS+) remote access method. Information in this chapter applies to all models of the CSS, except where noted.

This chapter contains the following major sections:

- Configuring the Secure Shell Daemon Protocol
- Configuring the CSS as a Client of a RADIUS Server
- Configuring the CSS as a Client of a TACACS+ Server
- Controlling Remote Access to the CSS
- Controlling Access to the CSS

Configuring the Secure Shell Daemon Protocol

The Secure Shell Daemon (SSHD) protocol provides secure encrypted communications between two hosts communicating over an insecure network. The CSS supports an implementation of OpenSSH to provide this secure communication. SSHD uses the standard CSS login sequence of entering the username and password at the CSS login prompts.

SSHD on the CSS supports both the SSH v1 and v2 protocols. For SSH v1, the software provides encrypted communication using ciphers such as 3DES or Blowfish. For SSH v2, the software provides 128-bit AES, Blowfish, 3DES, CAST128, Arcfour, 192-bit AES, or 256-bit AES.

Caution When using SSHD, ensure that the CSS is not configured to perform a network boot from a network-mounted file system on a remote system (a diskless environment). If you require the CSS to use the network-mounted method of booting, be aware that the SSHD protocol is not supported.

If the CSS has been booted using a network boot from a network-mounted file system, the CSS logs the following error message by SSHD as the protocol attempts to initialize (and then exit from operation):

```
Unable to initialize sshd; failure to seed random number generator
```

This section includes the following topics:

- Enabling SSH
- Configuring SSH Access
- Configuring SSHD in the CSS
- Configuring Telnet Access When Using SSHD
- Showing SSHD Configurations

Enabling SSH

To enable SSH functionality in your CSS, you must purchase the Secure Management software option. If you purchased the Secure Management software option:

- During the initial CSS order placement, the software Claim Certificate is included in the accessory kit.
- After receiving the CSS, Cisco Systems sends the Claim Certificate to you by mail.

Note If you cannot locate the Secure Management option Claim Certificate in the accessory kit, call the Cisco Technical Assistance Center (TAC) toll free, 24 hours a day, 7 days a week at 1-800-553-2447 or 1-408-526-7209. You can also e-mail TAC at tac@cisco.com.

Follow the instructions on the Claim Certificate to obtain the Secure Management software license key.

To enter the Secure Management license key and activate SSH:

1. Log in to the CSS and enter the **license** command.

```
# license
```

2. Enter the Secure Management license key.

```
Enter the Software License Key (q to quit): nnnnnnnnnnnn
```

The Secure Management license key is now properly installed and the SSH function activated.

Configuring SSH Access

SSH access to the CSS is enabled by default through the **no restrict ssh** command. You can verify the SSH access selection in the running-config file.

To enhance security when using SSHD, disable Telnet access (Telnet access is enabled by default). Use the **telnet-access disable** command as described in the "Controlling Access to the CSS" section.

To enable SSH access to the CSS, enter:

```
(config)# no restrict ssh
```

To disable SSH access, enter:

```
(config)# restrict ssh
```

Configuring SSHD in the CSS

The CSS provides the following commands for configuring SSHD:

- **sshd keepalive** - Enables TCP keepalive messages
- **sshd port** - Specifies the SSHD port
- **sshd server-keybits** - Sets the number of bits in the ephemeral protocol server key (SSH v1 only)

Ensure you enable SSHD access to the CSS for SSHD to accept connections from SSH clients. By default, SSH access is enabled through the **no restrict ssh** global command.

Configuring SSHD Keepalive

The CSS supports sending TCP keepalive messages to the client as a means for the server to determine whether the SSHD connection to the client is functioning (for example, if the network has gone down or the client has become unresponsive). If you disable sending SSHD keepalives to a client, sessions may hang indefinitely on the server, which consumes system resources.

Use the **sshd keepalive** command to enable SSHD keepalive. SSHD keepalive is enabled by default.

To enable sending SSHD keepalives to the client, enter:

```
(config)# sshd keepalive
```

To disable sending SSHD keepalives, enter:

```
(config)# no sshd keepalive
```

Configuring SSHD Port

Use the **sshd port** command to specify the port number to which the server listens for connections from clients. Enter a port number of 22 or from 512 to 65535. The default is 22 (the default port for SSH).

For example, to configure port number 65530 as the SSHD port, enter:

```
(config)# sshd port 65530
```

To reset the port number to the default of 22, enter:

```
(config)# no sshd port
```

Configuring SSHD Server-Keybits

Use the **sshd server keybits** command to specify the number of bits in the ephemeral protocol server key. The **sshd server keybits** command pertains only to SSH v1 connections. Enter the number of bits from 512 to 1024. The default is 768.

For example, to set the number of bits in the server key to 1024, enter:

```
(config)# sshd server-keybits 1024
```

To reset the number of bits to the default of 768, enter:

```
(config)# no sshd server-keybits
```

Configuring Telnet Access When Using SSHD

By default, Telnet access to the CSS is enabled. When you use SSHD, you can disable nonsecure Telnet access to the CSS. To enhance security when using SSHD, we recommend that you disable Telnet access. Use the global restrict telnet command to disable Telnet access to the CSS.

To disable Telnet access, enter:

```
(config)# restrict telnet
```

To reenable Telnet access to the CSS, enter:

```
(config)# no restrict telnet
```

Showing SSHD Configurations

Use the **show sshd** command to display SSHD configurations. This command provides the following options:

- **show sshd config** - Displays the SSHD configuration
- **show sshd sessions** - Displays a summary of the current active SSHD server sessions. The command displays data only if an SSH client is currently configured.
- **show sshd version** - Show the current version of the SSHield package running in the CSS.

To display the SSHD configuration, enter:

```
# show sshd config
```

Table 11-1 describes the fields in the **show sshd config** command output.

Table 11-1 Field Descriptions for the show sshd config Command

Field	Description
Maximum Sessions Allowed	The maximum number of concurrent SSHD sessions (five maximum).
Active Sessions	The number of currently active SSHD sessions.
Log Level	The current log level.
Listen Socket Count	The number of sockets that SSHD is currently listening on (not currently configurable, default is 1).
Listen Port	The port number that SSHD uses to listen for client connections (set by the **sshd port** command). The default is 22 (the default port for SSH). The port number is 22 or from 512 to 65535.

Table 11-1 Field Descriptions for the show sshd config Command (continued)

Field	Description
Listen Address	The address that SSHD uses to listen for client connections (not currently configurable; default is 0.0.0.0).
Server Key Bits	The number of bits to use when generating the SSHv1 server key. The default is 768. The range is from 512 to 1024.
RSA Protocol (SSH1)	The status of SSHv1 access (not currently configurable; default is enabled).
Empty Passwords	Disabled. The username must always have an associated password.
Keepalive	The status of sending a TCP keepalive to the client: Enabled or Disabled. SSHD keepalive is enabled by default.
SSH2 Cipher List	A list of SSHv2 cipher suites supported for authentication, encryption, and data integrity between the client and the server.

To display the SSHD sessions, enter:

```
# show sshd sessions
```

Table 11-2 describes the fields in the **show sshd sessions** command output.

Table 11-2 Field Descriptions for the show sshd sessions Command

Field	Description
Session_ID	The session ID.
Conn_TID	The connection task ID of the SSHD server handling the connection (tSshConn).
Login_TID	The login task ID handling the connection (tSshCli).

Table 11-2 Field Descriptions for the show sshd sessions Command (continued)

Field	Description
PTY_FD	The file descriptor used by the login task to communicate with the CSS CLI. The PTY_FD file descriptor allows you to correlate the SSH client sessions with those sessions listed under the Line field in the **show lines** output. For example, the **show sshd sessions** output displays an SSH client session connected to PTY_FD32. If you enter the **show lines** command you see a line in the display listing sshc32 (for SSH client pty_fd32). This correlation allows you to view the login time, idle time, and the location of the client of the SSH sessions through the **show lines** command.
Remote IP/ Remote Port	The remote IP and port number of the SSHD session.

To display the SSHD version, enter:

```
# show sshd version
SSHield version 1.5, SSH version OpenSSH_3.0.2p1
```

Configuring the CSS as a Client of a RADIUS Server

The Remote Authentication Dial-In User Service (RADIUS) protocol is a distributed client/server protocol that protects networks against unauthorized access. RADIUS uses the User Datagram Protocol (UDP) to exchange authentication and configuration information between the CSS authentication client and the active authentication server that contains all user authentication and network service access information. The RADIUS host is normally a multiuser system running RADIUS server software.

When a user remotely logs in to a CSS operating as a RADIUS client, the CSS sends an authentication request (including username, encrypted password, client IP address, and port ID) to the central RADIUS server. The RADIUS server is responsible for receiving user connection requests, authenticating users, and returning all configuration information necessary for the client to deliver services to the users. Transactions between the RADIUS client and the RADIUS server are authenticated through the use of a shared secret.

Once the RADIUS server receives the authentication request, it validates the sending client and consults a database of users to match the login request. If no response is returned by the RADIUS server within a period of time, the authentication request is retransmitted a predefined number of times. The RADIUS client can forward requests to an alternate secondary RADIUS server in the event that the primary server is down or is unreachable.

In a configuration where both a primary RADIUS server and a secondary RADIUS server are specified, and one or both of the RADIUS servers become unreachable, the CSS automatically transmits a keepalive authentication request to query the server(s). The CSS transmits the username “query” and the password “areyouup” to the RADIUS server (encrypted with the RADIUS server’s key) to determine the server’s state. The CSS continues to send this keepalive authentication request until the RADIUS server indicates it is available.

Use the **radius-server** command and its options to specify the RADIUS server host (primary RADIUS server, and, optionally, a secondary RADIUS server), communication time interval settings, and a shared secret text string. This command is available in global configuration mode.

This section includes the following topics:

- Configuring a RADIUS Server for Use with the CSS
- Specifying a Primary RADIUS Server
- Specifying a Secondary RADIUS Server
- Configuring the RADIUS Server Retransmits
- Configuring the RADIUS Server Dead-Time
- Showing RADIUS Server Configuration Information

After configuring the RADIUS server, enable RADIUS authentication for console and virtual logins (if the username and password pair is not in the local user database) through the **virtual authentication** and **console authentication** commands. See the "Controlling Remote Access to the CSS" section for details on the two commands.

Configuring a RADIUS Server for Use with the CSS

This section provides background information on the setup of a RADIUS server. It is intended as a guide to help ensure proper communication with a RADIUS server and a CSS operating as a RADIUS client.

The following sections summarize the recommended settings for the Cisco Secure Access Control Server (ACS) when used as a centralized RADIUS server with the CSS.

Configuring Authentication Settings

To configure the authentication settings on Cisco Secure ACS, go to the Network Configuration section of the Cisco Secure ACS HTML interface, the Add AAA Client page, and complete the following fields:

- AAA Client Hostname - Enter a name you want assigned to the CSS.
- AAA Client IP Address - Enter the IP address of the CSS Ethernet Management port or of a CSS circuit (depending on how the CSS is configured to communicate with the Cisco Secure ACS).

- Key - Enter the shared secret that the CSS and Cisco Secure ACS use to authenticate transactions. For correct operation, you must specify the identical shared secret on both the Cisco Secure ACS and the CSS. The key is case-sensitive.
- Authenticate Using - Select the **RADIUS (IETF)** network security protocol to use the standard IETF RADIUS attributes with the CSS.

Configuring Authorization Settings

To determine the privilege level of users accessing the CSS, you must configure the user accounts on the RADIUS server.

To configure the group authorization settings:

1. From the Group Setup section of the Cisco Secure ACS HTML interface, Group Setup Select page, select the group for which you want to configure RADIUS settings.
2. From the Group Settings section of the Cisco Secure ACS HTML interface, click the **IETF RADIUS Attributes, [006] Service-Type** checkbox. Then select **Administrative**. Administrative is required to enable RADIUS authentication for privileged user (SuperUser) connection with the CSS.

To add a user to a group, go to the **User Setup** section of the Cisco Secure ACS HTML interface:

- On the User Setup Select page, specify a username.
- On the User Setup Edit page, specify the following:
  - Password Authentication - Select an applicable authentication type from the list.
  - Password - Specify and confirm a password.
  - Group - Select the previously created RADIUS group to which you want to assign the user.

Specifying a Primary RADIUS Server

Use the **radius-server primary** command to specify a primary RADIUS server used to authenticate user information from the CSS RADIUS client (console or virtual authentication). The syntax for this global configuration mode command is:

radius-server primary *ip_address* **secret** *string* {**auth-port** *port_number*}

Options and variables for this command are as follows:

- **primary** *ip_address* - The IP address or host name for the primary RADIUS server. Enter the address in either dotted-decimal IP notation (for example, 192.168.11.1) or mnemonic host-name format (for example, myhost.mydomain.com).
- **secret** *string* - The shared secret text string between the primary RADIUS server and the CSS RADIUS client. The shared secret allows authentication transactions between the client and primary RADIUS server to occur. Enter the shared secret as a case-sensitive string with no spaces (16 characters maximum).
- **auth-port** *port_number* - (Optional) The UDP port on the primary RADIUS server allocated to receive authentication packets from the RADIUS client. Valid entries are 0 to 65535. The default is 1645.

To specify a primary RADIUS server, enter:

```
(config)# radius-server primary 172.27.56.76 secret Hello
auth-port 30658
```

To remove a primary RADIUS server, enter:

```
(config)# no radius-server primary
```

Specifying a Secondary RADIUS Server

Use the **radius-server secondary** command to specify a secondary RADIUS server to authenticate user information from the CSS RADIUS client (console or virtual authentication). The CSS directs authentication requests to the secondary RADIUS server when the specified RADIUS primary server is unavailable.

Note Configuration of a secondary RADIUS server is optional.

The syntax for this global configuration mode command is:

radius-server secondary *ip_address* **secret** *string* {**auth-port** *port_number*}

Options and variables for this command are as follows:

- **secondary** *ip_address* - The IP address or host name for the secondary RADIUS server. Enter the address in either dotted-decimal IP notation (for example, 192.168.11.1) or mnemonic host-name format (for example, myhost.mydomain.com).
- **secret** *string* - The shared secret text string between the secondary RADIUS server and the CSS RADIUS client. The shared secret allows authentication transactions between the client and secondary RADIUS server to occur. Enter the shared secret as a case-sensitive string with no spaces (16 characters maximum).
- **auth-port** *port_number* - (Optional) The UDP port on the primary RADIUS server allocated to receive authentication packets from the RADIUS client. Valid entries are 0 to 65535. The default is 1645.

To specify a secondary RADIUS server, enter:

```
(config)# radius-server secondary 172.27.56.79 secret Hello
auth-port 30658
```

To remove a secondary RADIUS server, enter:

```
(config)# no radius-server secondary
```

Configuring the RADIUS Server Timeouts

Use the **radius-server timeout** command to specify the time interval that the CSS waits for the RADIUS server (primary or secondary) to reply to an authentication request before retransmitting requests to the RADIUS server. You configure the number of retransmitted requests to the server through the **radius-server retransmit** command (see the "Configuring the RADIUS Server Retransmits" section). Valid entries are 1 to 255 seconds. The default is 10 seconds.

For example, to configure the configure the RADIUS server timeout interval to 1 minute (60 seconds), enter:

```
(config)# radius-server timeout 60
```

To reset the RADIUS server retransmit request to the default of 10 seconds, enter:

```
(config)# no radius-server timeout
```

Configuring the RADIUS Server Retransmits

Use the **radius-server retransmit** command to specify the number of times the CSS retransmits an authentication request to a timed-out RADIUS server before considering the server dead and stopping transmission. If a secondary RADIUS server has been identified, the server is selected as the active server. Valid entries are 1 to 30 retries. The default is 3 retransmissions.

If the RADIUS server does not respond to the CSS retransmitted requests, the CSS considers the server as dead, stops transmitting to the server, and starts the dead timer as defined through the **radius-server dead-time** command (see the "Configuring the RADIUS Server Dead-Time" section). If a secondary server is configured, the CSS transmits the requests to the secondary server. If the secondary server does not respond to the request, the CSS considers the server dead and starts the dead timer. If there is no active server, the CSS stops transmitting requests until the primary RADIUS server becomes alive.

For example, to configure the number of RADIUS server retransmissions to 5, enter:

```
(config)# radius-server retransmit 5
```

To reset the RADIUS server retransmit request to the default of 3 retransmissions, enter:

```
(config)# no radius-server retransmit
```

Configuring the RADIUS Server Dead-Time

Use the **radius-server dead-time** command to set the time interval in which the CSS verifies whether a nonresponsive server is operational. During the set time interval, the CSS sends probe access-request packets to verify that the RADIUS server (primary or secondary) is available and can receive authentication requests. The dead-time interval starts when the server does not respond to the number of authentication request transmissions configured through the **radius-server retransmit** command. When the server responds to a probe access-request packet, the CSS transmits the authentication request to the server. Valid entries are 1 to 255 seconds. The default is 5 seconds.

To configure the RADIUS server dead-time to 15 seconds, with probe access-requests enabled, enter:

```
(config)# radius-server dead-time 15
```

To reset the RADIUS server dead-time request to the default of 5 seconds, enter:

```
(config)# no radius-server dead-time
```

Showing RADIUS Server Configuration Information

Use the **show radius** command to display information and statistics about the RADIUS server configuration. The syntax and options for the command are as follows:

- **show radius config** [**primary**|**secondary**|**all**] - Displays RADIUS configuration information for a specific server or all servers, identified by type
- **show radius stat** [**primary**|**secondary**|**all**] - Displays RADIUS authentication statistics for a specific server or all servers, identified by type

To view the configuration for a RADIUS primary server, enter:

```
(config)# show radius config primary
```

To view the authentication statistics for a RADIUS secondary server, enter:

```
(config)# show radius stats secondary
```

Table 11-3 describes the fields in the **show radius config** command output.

Table 11-3 Field Descriptions for the show radius config Command

Field	Description
Server IP Address	The IP address or host name for the specified RADIUS server
Secret	The shared secret text string between the specified RADIUS server and the CSS RADIUS client
Port	The UDP port on the specified RADIUS server allocated to receive authentication packets from the CSS RADIUS client; the default port number is 1645
State	The operational stats of the RADIUS server (ALIVE, DOWN, UNKNOWN)
Dead Timer	The time interval (in seconds) that the CSS probes a nonresponsive RADIUS server (primary or secondary) to determine whether it is operational and can receive authentication requests
Timeout	The interval (in seconds) that the CSS RADIUS client waits for the RADIUS server to reply to an authentication request before retransmitting requests to the RADIUS server
Retransmit Limit	The number of times the CSS RADIUS client retransmits an authentication request to a timed out RADIUS server before stopping transmission to that server
Probes	The packets that the CSS RADIUS client automatically transmits as a means to determine whether the RADIUS server is still available and can receive authentication requests

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

Table 11-4 Field Descriptions for the show radius stat Command

Field	Description
Server IP address	The IP address or host name of the specified RADIUS server
Accepts	The number of times the RADIUS server accepts an authentication request from the CSS RADIUS client

Table 11-4 Field Descriptions for the show radius stat Command (continued)

Field	Description
Requests	The number of times the CSS RADIUS client issues an authentication request to the RADIUS server
Retransmits	The number of times the CSS RADIUS client retransmits an authentication request to the active RADIUS server after a timeout occurred
Rejects	The number of times the CSS RADIUS client receives a reject notification from the RADIUS server while trying to establish an authentication request
Bad Responses	The number of times the CSS RADIUS client receives a bad transmission from the RADIUS server
Bad Authenticators	The number of times the RADIUS server denies an authentication request from the CSS RADIUS client
Pending Requests	The number of pending authentication requests to the RADIUS server
Timeouts	The number of times the CSS RADIUS client reached the specified timeout interval while waiting for the RADIUS server to reply to an authentication request
Discarded Authentication Requests	The number of authentication requests that were discarded while the primary or secondary RADIUS server was down

Configuring the CSS as a Client of a TACACS+ Server

The Terminal Access Controller Access Control System (TACACS+) protocol provides access control for routers, network access servers (NAS), or other devices through one or more daemon servers. TACACS+ encrypts all traffic between the NAS and daemon using TCP communications for reliable delivery.

You can configure the CSS as a client of a TACACS+ server to provide a method for authentication of users, and a method of authorization and accounting of configuration and nonconfiguration commands.

This section includes the following topics:

- Configuring TACACS+ Server User Accounts for Use with the CSS
- Defining a TACACS+ Server
- Defining a Global Encryption Key
- Setting the Global CSS TACACS+ Timeout Period
- Setting TACACS+ Authorization
- Setting TACACS+ Accounting
- Showing TACACS+ Server Configuration Information

After you configure the TACACS+ server on the CSS, configure TACACS+ authentication for virtual or console authentication. See the "Controlling Remote Access to the CSS" section for details.

Configuring TACACS+ Server User Accounts for Use with the CSS

This section provides background information on the setup of a TACACS+ server. It is intended as a guide to help ensure proper communication with a TACACS+ server and a CSS operating as a TACACS+ client.

The following sections summarize the recommended Cisco Secure Access Control Server (ACS) TACACS+ user authentication and authorization settings.

Configuring Authentication Settings

To configure the authentication settings on Cisco Secure ACS, go to the Network Configuration section of the Cisco Secure ACS HTML interface, the Add AAA Client page, and complete the following fields:

- AAA Client Hostname - Enter a name you want assigned to the CSS.
- AAA Client IP Address - Enter the IP address of the CSS Ethernet management port or of a CSS circuit (depending on how the CSS is configured to communicate with the Cisco Secure ACS).
- Key - Enter the shared secret that the CSS and Cisco Secure ACS use to authenticate transactions. For correct operation, you must specify the identical shared secret on both the Cisco Secure ACS and the CSS. The key is case-sensitive.
- Authenticate Using - Select **TACACS+ (Cisco IOS)**.

Configuring Authorization Settings

To determine the privilege level of users accessing the CSS, you must configure the user accounts on the TACACS+ server to permit or deny execution of the **privilege** command. The CSS queries the TACACS+ server for authorization to execute the **privilege** command. If the server allows the **privilege** command, the user is granted privileged (SuperUser and configuration modes) access to the CSS. If the server denies the **privilege** command, the user is granted nonprivileged (User mode) access to the CSS.

To configure the group authorization settings:

1. From the Group Setup section of the Cisco Secure ACS HTML interface, Group Setup Select page, select the group for which you want to configure TACACS+ settings.
2. On the Shell Command Authorization Set page, click the **Per Group Command Authorization** checkbox

3. Under **Unmatched Cisco IOS Commands**, either permit or deny execution of the privilege command:
   - For a group that has SuperUser privileges on the CSS, select **Permit**. A SuperUser can issue any CSS command.
   - For a group that has User privileges on the CSS, select **Deny**. A user can issue CSS commands that does not change the CSS configuration; for example, **show** commands.

An alternative way to configure the group authorization settings is as follows:

1. Select **Shared Profile Components**, **Shell Command Authorization Sets** page.
2. Click the **Add** button to add a set or to edit an existing set.
3. Enter a name and description.
4. Proceed next to Unmatched Commands, either permit or deny execution of the privilege command:
   - For a user that has SuperUser privileges on the CSS, click **Permit**. A SuperUser can issue any CSS command.
   - For a user that has User privileges on the CSS, click **Deny**. A user can issue CSS commands that do not change the CSS configuration; for example, **show** commands.
5. From the Group Setup section, Group Setup Select page, select the group for which you want to configure TACACS+ settings.
6. On the Shell Command Authorization Set section, select **Assign a Shell Command Authorization Set for any network device**.
7. Select the set from the list.

To add a user to a group, go to the **User Setup** section of the Cisco Secure ACS HTML interface:

- On the User Setup Select page, specify a username.
- On the User Setup Edit page, specify the following:
  - Password Authentication - Select an applicable authentication type from the list.
  - Password - Specify and confirm a password.
  - Group - Select the previously created TACACS+ group to which you want to assign the user.

Defining a TACACS+ Server

The TACACS+ server contains the TACACS+ authentication, authorization, and accounting databases. You can designate a maximum of three servers on the CSS. However, the CSS uses only one server at a time. The CSS selects the server based upon availability, giving preference to the configured primary server. The CSS sends periodic TCP keepalive probes at a frequency of every five seconds to the TACACS+ server to determine its operational state: Alive, Dying, or Dead. The TCP keepalive frequency is not user-configurable in the CSS.

Note For general guidelines on the recommended setup of a TACACS+ server (the Cisco Secure Access Control Server in this example), see the "Configuring TACACS+ Server User Accounts for Use with the CSS" section.

To apply a TACACS+ global attribute, such as the timeout period or shared secret, to a TACACS+ server, you must configure the global attribute before you configure the server. To apply a modified global attribute to a configured CSS TACACS+ server, remove the server and reconfigure it.

Use the **tacacs-server** command to define a server. You must provide the IP address and port number for the server. You can optionally define the timeout period and encryption key and designate the server as the primary server.

The syntax for this global configuration command is:

tacacs-server *ip_address port* {*timeout* [**"***cleartext_key***"**|*des_key*]} {**primary**}

The variables and options for this command are as follows:

- *ip_address* - The IP address of the TACACS+ server. Enter the IP address in dotted-decimal format.
- *port* - The TCP port of TACACS+ server. The default port is 49. You can enter a port number from 1 to 65535.
- *timeout* - (Optional) The amount of time to wait for a response from the server. Enter a number from 1 to 255. The default is 5 seconds. Defining this option overrides the **tacacs-server timeout** command. For more information on the TACACS+ timeout period and setting a global timeout, see the "Setting the Global CSS TACACS+ Timeout Period" section.

- **"***cleartext_key***"**|*des_key* - (Optional) The shared secret between the CSS and the server. You must define an encryption key to encrypt TACACS+ packet transactions between the CSS and the TACACS+ server. If you do not define an encryption key, packets are not encrypted.

  The shared secret value is identical to the one on the TACACS+ server. The shared secret key can be either clear text entered in quotes or the DES-encrypted secret entered without quotes. The clear text key is DES-encrypted before it is placed in the running configuration. Either key type can have a maximum of 100 characters.

  Defining this option overrides the **tacacs-server key** command. For more information on defining a global encryption key, see the "Defining a Global Encryption Key" section.

- **primary** - (Optional) Assigns the TACACS+ server precedence over the other configured servers. You can specify only one primary server.

Note If you need to change a timeout period or the shared secret for a specific server, you must delete the server and redefine it with the updated parameter.

For example, to define a primary TACACS+ server at IP address 192.168.11.1 with a default port of 49, a timeout period of 12 seconds, and a clear text shared secret of summary, enter:

```
#(config) tacacs-server 192.168.11.1 12 "summary" primary
```

To delete a TACACS+ server at IP address 192.168.11.1 with a default port of 49, enter:

```
#(config) no tacacs-server 192.168.11.1 49
```

After configuring the TACACS+ server, enable TACACS+ authentication for console and virtual logins (if the username and password pair is not in the local user database) through the **virtual authentication** and **console authentication** commands. See the "Controlling Remote Access to the CSS" section for information about the two commands.

Defining a Global Encryption Key

The CSS allows you to define a global encryption key for communications with all configured TACACS+ servers. To encrypt TACACS+ packet transactions between the CSS and the TACACS+ server, you must define an encryption key. If you do not define an encryption key, packets are not encrypted. The key is a shared secret value that is identical to the one on the TACACS+ server. Use the **tacacs-server key** command to specify a shared secret between the CSS and the server.

A shared secret defined when specifying a TACACS+ server overrides the global secret (see the "Defining a TACACS+ Server" section). To apply a modified global shared secret to a configured CSS TACACS+ server, remove the server and reconfigure it.

The shared secret key can be either clear text entered in quotes or the DES-encrypted secret. The clear text key is DES-encrypted before it is placed in the running configuration. Either key types can have a maximum of 100 characters.

For example, to define the clear text key, enter:

```
#(config) tacacs-server key "market"
```

To define a DES-encrypted key, enter:

```
#(config) tacacs-server key acskefterefesdtx
```

To remove the key, enter:

```
#(config) no tacacs-server key
```

Setting the Global CSS TACACS+ Timeout Period

The CSS allows you to define a global TACACS+ timeout period for use with all configured TACACS+ servers. To determine the availability of the TACACS+ servers, the CSS sends periodic TCP keepalive probes to them. If the server does not respond to the probe within the timeout period, the CSS considers the server unavailable.

If the CSS attempts to contact the server and does not receive a response within the defined timeout value, it uses another server. The next configured server is contacted and the process repeated. If a second (or third) TACACS+ server has been identified, the CSS selects that server as the active server.

If the CSS cannot reach all three TACACS+ servers, users are not authenticated and cannot log in to the CSS unless TACACS+ is used in combination with a RADIUS or local server, as defined through the **virtual** command or the **console** command. See the "Controlling Remote Access to the CSS" section for details about the two commands.

By default, the global timeout period is 5 seconds. Use the **tacacs-server timeout** command to change the timeout period. Enter a number from 1 to 255.

The timeout period defined when specifying a TACACS+ server overrides the global timeout period (see the "Defining a TACACS+ Server" section). To apply a modified global timeout period to a configured CSS TACACS+ server, remove the server and reconfigure it.

For example, to set the timeout period to 60 seconds, enter:

```
#(config) tacacs-server timeout 60
```

To reset the timeout period to the default of 5 seconds, enter:

```
#(config) no tacacs-server timeout
```

Setting TACACS+ Authorization

TACACS+ authorization allows the TACACS+ server to control specific CSS commands that the user can execute. CSS authorization divides the command set into two categories:

- Configuration commands that change the CSS running configuration.
- Nonconfiguration commands that do not change the running configuration. These commands include, but are not limited to, mode transition, show, and administrative commands.

By default, authorization is disabled. When authorization is enabled, the TACACS+ server is responsible for granting permission or denying all attempts to issue commands.

When you enable authorization, the exchange between the TACACS+ server and the CSS causes a delay in executing the command. Failure of the TACACS+ server results in the failure of all authorization requests and the suspension of user activity unless another server is reachable. To enable users to execute commands in this case, configure a failover authentication method to a local user database. Users must log back in to the CSS.

Use the **tacacs-server authorize config** command to enable authorization of all commands that change the running configuration. For example:

```
#(config) tacacs-server authorize config
```

Use the **tacacs-server authorize non-config** command to enable authorization of all commands that do not change the running configuration. For example:

```
#(config) tacacs-server authorize non-config
```

Use the **no** form of these commands to disable authorization. For example, to disable authorization for commands that affect the running configuration, enter:

```
#(config) no tacacs-server authorize config
```

To disable authorization for commands that do not affect the running configuration, enter:

```
#(config) no tacacs-server authorize non-config
```

Setting TACACS+ Accounting

TACACS+ accounting allows the TACACS+ server to receive an accounting report for commands that the user can execute. CSS accounting divides the command set into two categories:

- Configuration commands that change the CSS running configuration.
- Nonconfiguration commands that do not change the running configuration. These commands include, but are not limited to, mode transition commands, show commands, and administrative commands.

By default, the CSS disables accounting. When you enable accounting, you can account for configuration commands, nonconfiguration commands, or both.

Note Failure of the TACACS+ server does not result in the suspension of user activity.

Use the **tacacs-server account config** command to enable the TACACS+ server to receive accounting reports for all commands that change the running configuration. For example:

```
#(config) tacacs-server account config
```

Use the **tacacs-server account non-config** command to enable the TACACS+ server to receive accounting reports for all commands that do not change the running configuration. For example:

```
#(config) tacacs-server account non-config
```

Use the **no** form of these commands to disable accounting. For example, to disable accounting for commands that affect the running configuration, enter:

```
#(config) no tacacs-server account config
```

To disable accounting for commands that do not affect the running configuration, enter:

```
#(config) no tacacs-server account non-config
```

Showing TACACS+ Server Configuration Information

Use the **show tacacs-server** command to display the TACACS+ server configuration information. To view this information, enter:

```
(config)# show tacacs-server
```

Table 11-5 describes the fields in the **show tacacs-server** command output.

Table 11-5 Field Descriptions for the show tacacs-server Command

Field	Description
IP/Port	The TACACS+ server IP address and port number
State	The operational state of the server (Alive, Dying, or Dead) determined by the internal TCP Keepalive
Primary	Indicates whether this record is the primary TACACS+ server
Authen	The number of authentication requests made to the TACACS+ server
Author	The number of authorization requests made to the TACACS+ server
Account	The number of accounting requests made to the TACACS+ server
Global Timeout	How long the CSS waits for a response from a TACACS+ server
Global Key	Shared secret, used by all TACACS+ servers, unless individually configured for the server
Authorize Config Commands	Indicates whether configuration commands receive authorization
Authorize Non-Config	Indicates whether nonconfiguration commands receive authorization

Controlling Remote Access to the CSS

To control access to the CSS, you can configure the CSS to authenticate remote (virtual) or console users. The CSS can authenticate users by using the local user database, RADIUS server, or TACACS+ server. You can also allow user access without authenticating or disallowing all remote user access to the CSS.

You can set a maximum of three authentication methods: a primary, secondary, or tertiary authentication method. The primary method is the first authentication method that the CSS tries. If the primary authentication method fails (for example, the RADIUS server is down or is unreachable), the CSS tries the secondary method. And if the secondary method fails, then the CSS tries the tertiary method. In the event the tertiary method also fails, the CSS displays a message that authentication has failed.

The CSS does not attempt a secondary or tertiary authentication method under the following conditions:

- If the authentication method is **local**, and the local username is not found in the local user database.
- If the authentication method is **local** and the local username is found in the local user database, but the password is invalid.
- If the authentication method is **radius**, and the RADIUS server rejects the primary authentication request from the CSS.
- If the authentication method is **tacacs**, and the TACACS+ server rejects the primary authentication request from the CSS.

Before you can use RADIUS or TACACS+ as either the virtual authentication method or the console authentication method, you must enable communication with the RADIUS or TACACS+ security server. Use either the **radius-server** command (see the "Configuring the CSS as a Client of a RADIUS Server" section) or the **tacacs-server** command (see the "Configuring the CSS as a Client of a TACACS+ Server" section).

This section includes the following topics:

- Configuring Virtual Authentication
- Configuring Console Authentication

To display virtual and console authentication settings, use the **show user-database** command (refer to Chapter 3, Managing the CSS Software).

Configuring Virtual Authentication

Virtual authentication allows remote users to log in to the CSS when they are using FTP, Telnet, SSHD, or the Device Management user interface with or without requiring a username and password. The CSS can also deny access to all remote users.

You can configure the CSS to authenticate users by using the local database, RADIUS server, or TACACS+ server. By default, the CSS uses the local database as the primary method to authenticate users and disallows user access for the secondary and tertiary method.

Use the **virtual authentication** command to configure the primary, secondary, or tertiary virtual authentication method. The syntax for this global configuration command is:

virtual authentication [**primary**|**secondary**|**tertiary**
[**local**|**radius**|**tacacs**|**disallowed**]]

The options for this command are as follows:

- **primary** - Defines the first authentication method that the CSS uses. The default primary virtual authentication method is the local user database.
- **secondary** - Defines the second authentication method that the CSS uses if the first method fails. The default secondary virtual authentication method is to disallow all user access.

Note If you are configuring a TACACS+ server as the primary authentication method, define a secondary authentication method, such as **local**.

- **tertiary** - Defines the third authentication method that the CSS uses if the second method fails. The default tertiary virtual authentication method is to disallow all user access.
- **local** - The CSS uses the local user database for authentication.
- **radius** - The CSS uses the configured RADIUS server for authentication.
- **tacacs** - The CSS uses the configured TACACS+ server for authentication.
- **disallowed** - The CSS disallows access by all remote users. Entering this option does not terminate existing connections.

To remove users currently logged in to the CSS, use the **disconnect** command.

To define the TACACS+ server as the primary virtual authentication method, enter:

```
#(config) virtual authentication primary tacacs
```

To define local user database as the secondary virtual authentication method, enter:

```
#(config) virtual authentication secondary local
```

Configuring Console Authentication

Console authentication allows users to log in to the CSS through a terminal connected to the console port with or without requiring a username and password. The CSS cannot disallow user access as a primary authentication method; however, it can disallow user access as a secondary or tertiary authentication method.

You can configure the CSS to authenticate users by using the local database, RADIUS server, or TACACS+ server. By default, the CSS uses the local database as the primary method to authenticate users and disallows user access for the secondary and tertiary method.

Use the **console authentication** command to configure the primary, secondary, or tertiary console authentication method. The syntax for this global configuration command is:

console authentication [**primary** [**local**|**radius**|**tacacs**|**none**]
    |**secondary**|**tertiary** [**local**|**radius**|**tacacs**|**none**|**disallowed**]]

The options for this command are as follows:

- **primary** - Defines the first authentication method that the CSS uses. The default primary console authentication method is the local user database.
- **local** - The CSS uses the local user database for authentication.
- **radius** - The CSS uses the configured RADIUS server for authentication.
- **tacacs** - The CSS uses the configured TACACS+ server for authentication.
- **none** - The CSS uses no authentication method. All users can access the CSS.
- **secondary** - Defines the second authentication method that the CSS uses if the first method fails. The default secondary console authentication method is to disallow all user access.

Note If you are configuring a TACACS+ server as the primary authentication method, define a secondary authentication method, such as **local**. If you do not configure a secondary method and use the default of **disallowed**, you have the possibility of being locked out of the CSS.

- **tertiary** - Defines the third authentication method that the CSS uses if the second method fails. The default tertiary console authentication method is to disallow all user access.
- **disallowed** - The CSS disallows access by all users (secondary or tertiary authentication method only). Entering this option does not terminate existing connections.

To remove users currently logged in to the CSS, use the **disconnect** command.

To define the TACACS+ server as the primary console authentication method, enter:

```
#(config) console authentication primary tacacs
```

To define local user database as the secondary console authentication method, enter:

```
#(config) console authentication secondary local
```

To disable authentication on the console port allowing users to access the CSS without a username and password, enter:

```
#(config) no console authentication
```

Controlling Access to the CSS

Use the **restrict** and **no restrict** commands to enable or disable console, FTP, SNMP, SSH, Telnet, user database, XML, and web management data transfer to the CSS. CSS access through a console, FTP, SSHD, SNMP, and Telnet is enabled by default. The CSS supports a maximum of four FTP sessions and a maximum of four Telnet sessions.

Specifying the **restrict** command does not prevent the CSS from listening for connection attempts on the restricted port. For TCP connections, the CSS completes the TCP 3-way handshake, then terminates the connection with an error to prevent any data transfer from occurring. For UDP SNMP connections, the CSS simply discards the packets.

To secure restricted ports from unauthorized access, configure ACL clauses to deny packets destined to these ports, while permitting normal traffic to flow through the CSS. You can also use ACLs to secure the CSS itself. Refer to the *Cisco Content Services Switch Basic Configuration Guide* for information about configuring ACLs for the CSS.

Enabling Access to the CSS

To enable console, FTP, SNMP, SSH, Telnet, user database, XML, and web management access to the CSS, use the following **no restrict** commands:

- **no restrict console** - Enables console access to the CSS (enabled by default)
- **no restrict ftp** - Enables FTP access to the CSS (enabled by default)
- **no restrict ssh** - Enables SSHD access to the CSS (enabled by default)
- **no restrict snmp** - Enables SNMP access to the CSS (enabled by default)
- **no restrict telnet** - Enables Telnet access to the CSS (enabled by default)
- **no restrict user-database** - Enables users to clear the running-config file and create or modify usernames. Only administrator and technician users can perform these tasks (enabled by default).
- **no restrict xml** - Enables XML access to the CSS (disabled by default)
- **no restrict web-mgmt** - Enables Device Management user interface access to the CSS (disabled by default)

Note Disable Telnet access when you want to use the Secure Shell Host (SSH) server. For information about configuring SSH, see the "Configuring the Secure Shell Daemon Protocol" section.

For example, to enable Device Management user interface access, enter:

```
(config)# no restrict web-mgmt
```

Refer to Chapter 12, Configuring Simple Network Management Protocol (SNMP) for details on configuring the Simple Network Management Protocol (SNMP) features on your CSS. For details on making web-based configuration changes to the CSS using Extensible Markup Language (XML), refer to the *Cisco Content Services Switch Advanced Configuration Guide*. For details on using the Device Management user interface, refer to the *Cisco Content Services Switch Device Management User's Guide*.

Disabling Access to the CSS

To disable console, FTP, SNMP, SSH, Telnet, user database, XML, and web management access to the CSS, use the following **restrict** commands:

- **restrict console** - Disables console access to the CSS (enabled by default)
- **restrict ftp** - Disables FTP access to the CSS (enabled by default)
- **restrict snmp** - Disables SNMP access to the CSS (enabled by default)
- **restrict ssh** - Disables SSHD access to the CSS (enabled by default)
- **restrict telnet** - Disables Telnet access to the CSS (enabled by default)
- **restrict user-database** - Prevents users from clearing the running-config file and creating or modifying usernames. Only administrator and technician users can perform these tasks (enabled by default).
- **restrict xml** - Disables XML access to the CSS (disabled by default)
- **restrict web-mgmt** - Disables web management access to the CSS (disabled by default)

For example, to disable Telnet access, enter:

```
(config)# restrict telnet
```

Where to Go Next

Chapter 12, Configuring Simple Network Management Protocol (SNMP) describes how to configure SNMP on the CSS.

Configuring Simple Network Management Protocol (SNMP)

This chapter provides information on configuring Simple Network Management Protocol (SNMP) features of your CSS. It also provides a brief overview of SNMP, an Application Layer protocol used extensively in the communications industry. Information in this chapter applies to all CSS models except where noted.

This chapter includes the following major sections:

- SNMP Overview
- Management Information Base (MIB) Overview
- Preparing to Configure SNMP on the CSS
- Defining the CSS as an SNMP Agent
- Configuring Denial of Service (DoS)
- Displaying the SNMP Configuration
- Managing SNMP on the CSS
- CSS SNMP Traps
- CSS MIBs

SNMP Overview

SNMP is a set of network management standards for IP-based internetworks. SNMP includes a protocol, a database-structure specification, and a set of management data objects. SNMP implementations typically consist of a management application running on one or more network management systems (NMSs), and agent applications, usually executing in firmware on various network devices.

SNMP has two major standard revisions, SNMPv1 and SNMPv2. The CSS supports both SNMPv1 and SNMPv2C (SNMP version 2C), a standard Management Information Base (MIB-II) object, along with an extensive set of enterprise MIB objects. MIBs are discussed in the "Management Information Base (MIB) Overview" section.

This section contains the following topics:

- Managers and Agents
- SNMP Manager and Agent Communication

Note By default, SNMP access to the CSS is enabled through the **no restrict snmp** command. For details, see the "Preparing to Configure SNMP on the CSS" section.

Managers and Agents

SNMP uses software entities called *managers* and *agents* to manage network devices:

- The *manager* monitors and controls all other SNMP-managed devices (network nodes) in the network. There must be at least one SNMP manager in a managed network. The manager is installed on a workstation somewhere in the network.
- An *agent* resides in a managed device (a network node). The agent receives instructions from the SNMP manager, and also sends management information back to the SNMP manager as events occur. The agent can reside on routers, bridges, hubs, workstations, or printers, to name just a few network devices.

There are many different SNMP management applications, but they all perform the same basic task: They allow SNMP managers to communicate with agents to monitor, configure, and receive alerts from the network devices. You can use any SNMP-compatible NMS to monitor and control a CSS.

SNMP Manager and Agent Communication

There are several ways that the SNMP manager and the agent communicate.

- The manager can:
  - Retrieve a value (a GET action).

    The SNMP manager requests information from the agent, such as the number of users logged on to the agent device, or the status of a critical process on that device. The agent gets the value of the requested MIB object and sends the value back to the manager.
  - Retrieve the value immediately after the variable you name (a GET-NEXT action).

    The SNMP manager retrieves values from within a MIB. Using the get-next function, you do not need to know the exact MIB object instance you are looking for; the SNMP manager takes the variable you name and then uses a sequential search to find the desired variables.

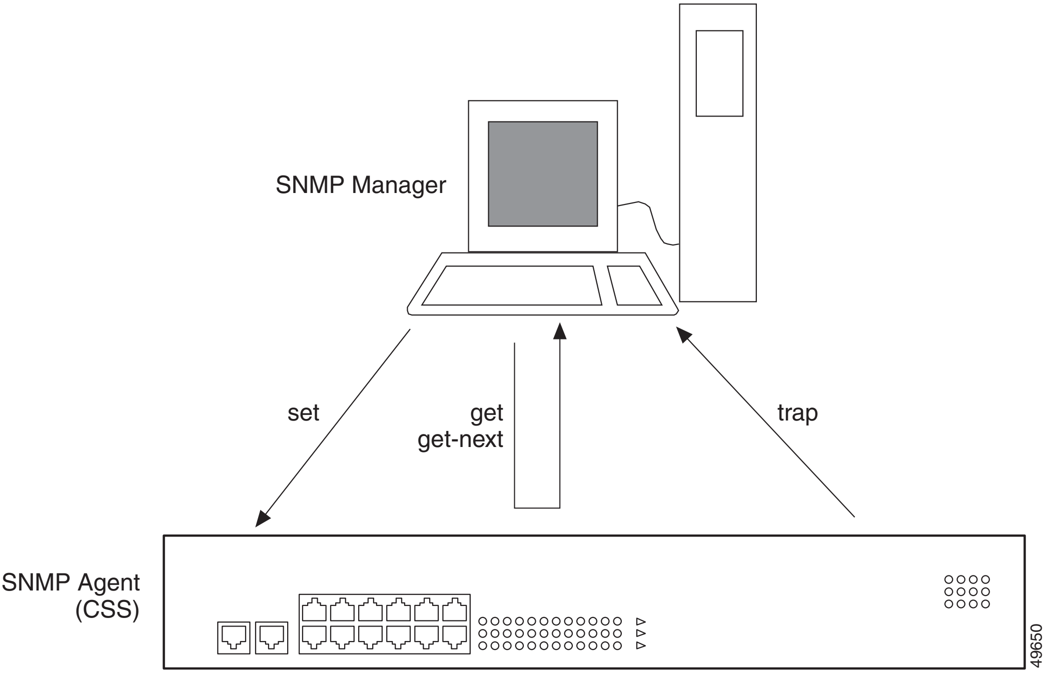
  - Retrieve a number of values (a GET-BULK action).

    The SNMP manager performs a number of get-next actions that you specify.
  - Change a setting on the agent (a SET action).

    The SNMP manager requests the agent to change the value of the MIB object. For example, you could run a script or an application on a remote device with a set action.
- An agent can send an unsolicited message to the manager at any time if a significant, predetermined event takes place on the agent. This message is called a *trap*. For details on SNMP traps (and associated MIB objects) supported by the CSS software, see the "CSS SNMP Traps" section.

  When a trap condition occurs, the SNMP agent sends an SNMP trap message to the device specified as the *trap receiver* or *trap host*. The SNMP Administrator configures the trap host (usually the SNMP management station) to perform the action needed when a trap is detected. Figure 12-1 illustrates SNMP manager and agent communication.

Figure 12-1 SNMP Manager and Agent Interaction



Management Information Base (MIB) Overview

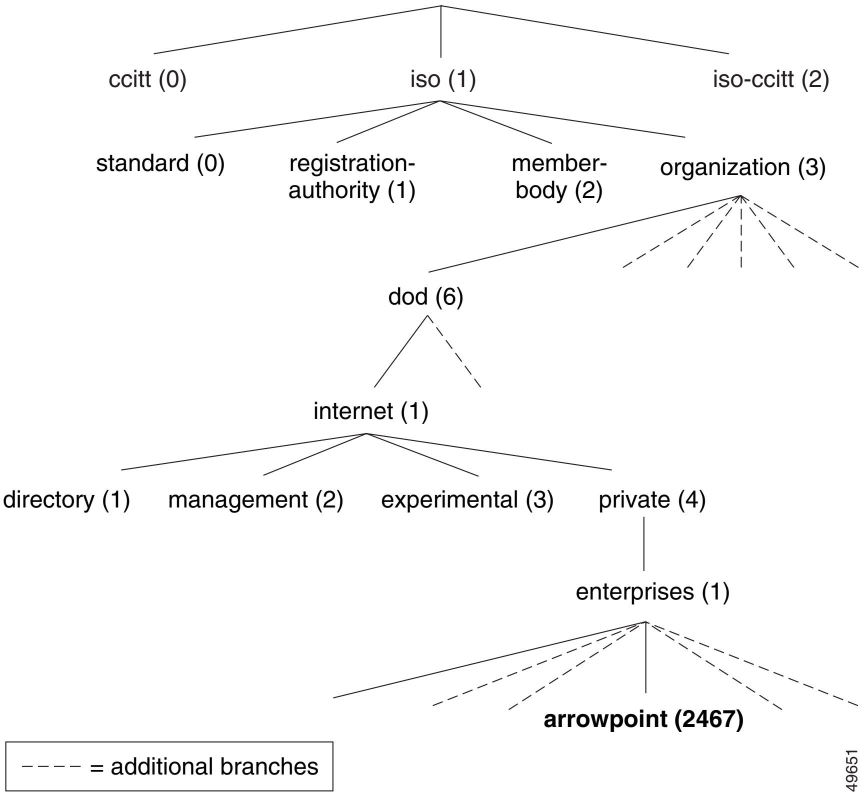
SNMP obtains information from the network through a Management Information Base (MIB). The MIB is a database of code blocks called *MIB objects*. Each MIB object controls one specific function, such as counting how many bytes are transmitted through an agent's port. The MIB object comprises *MIB variables*, which define the MIB object name, description, default value, and so forth.

The collection of MIB objects is structured hierarchically. The MIB hierarchy is referred to as the *MIB tree*. The MIB tree is defined by the International Standards Organization (ISO). The MIB is installed on the SNMP manager and is present within each agent in the SNMP network.

At the top of the tree is the broadest information about a network. Each branch and sub-branch of the tree gets progressively more specific, and the lowest branches of the tree contain the most specific MIB objects; the leaves contain the actual data. Figure 12-2 shows an example of how the MIB tree objects become more specific as the tree expands.

Note There are two versions of the MIB tree as defined by ISO: MIB-I and MIB-II. MIBII has more variables than MIB-I. Refer to the MIB-II standard in RFC 1213, "Management Information Base for Network Management of TCP/IP-based Internets: MIB-II."

Figure 12-2 Top of the MIB Tree



This section includes the following topics:

- MIB Variables
- MIB Extensions (Enterprise MIBs)
- Updating MIB Files

MIB Variables

There are two types of MIB variables:

- Scalar - Variables that define an object with a single representation. This means that the object describes a particular characteristic of the entire system. An example of a scalar variable is **SysDescr**, which provides a system-wide description of the CSS.
- Tabular - Variables that define an object with multiple representations. This means that the object can have different values, depending on the qualifier. For example, one tabular object could show bytes per interface, temperature per board, or hits per service.

As shown in Figure 12-2, a number is associated with a MIB object name. This number is called the *object identifier* (or *object ID*), and it uniquely identifies the MIB object in the MIB tree. (The dotted lines represent other branches not relevant to this discussion.)

For example, note in Figure 12-2 that the MIB object labeled *arrowpoint (2467)*, which contains the MIB objects specific to the CSS, can be labeled:

```
iso.organization.dod.internet.private.enterprises.arrowpoint
```

or

```
1.3.6.1.4.1.2467
```

MIB Extensions (Enterprise MIBs)

The MIB tree has a special branch set aside for specific vendors to build their own extensions; this special branch is called the *Enterprise MIB branch*. The CSS MIBs are included in the CSS GZIP file and are located in the CSS /mibs directory. The MIB files in this branch comprise the CSS Enterprise MIBs (the highlighted MIB identifier in Figure 12-2). The enterprise MIB files are categorized along functional boundaries.

For a list of MIB branches under the CSS Enterprise MIB, see the "CSS MIBs" section.

Updating MIB Files

We recommend that you update the CSS Enterprise MIBs after you upgrade the CSS software. CSS MIBs are included in the CSS GZIP file. During the software upgrade, the MIBs are loaded into the CSS /mibs directory.

To update the CSS MIBs on your management station after you upgrade the CSS:

1. Transfer the MIBs using FTP from the CSS MIBs (/v1 or /v2) directory to your management station.
2. Load the MIBs into the management application.

SNMP Communities

Each SNMP device or member is part of a *community*. An SNMP community determines the access rights for each SNMP device.

You supply a name to the community. After that, all SNMP devices that are assigned to that community as *members* have the same access rights. The access rights that the CSS supports are:

- read - Allows read-only access to the MIB tree for devices included in this community
- read-write - Allows both read and write access to the MIB tree for devices included in this community

Preparing to Configure SNMP on the CSS

Once you have set up your SNMP management application, you are ready to configure SNMP settings on the CSS. You can configure two basic areas of SNMP functionality on the CSS: SNMP functions and RMON functions.

Note Refer to Chapter 13, Configuring Remote Monitoring (RMON) for information on configuring RMON.

To control SNMP access to the CSS, use the **no restrict snmp** and **restrict snmp** commands. Access through SNMP is enabled by default. The options for this global configuration mode command are:

- **no restrict snmp** - Enables SNMP access to the CSS (default setting)
- **restrict snmp** - Disables SNMP access to the CSS

Before you set up SNMP on your network consider the following items when planning your SNMP configuration:

- Decide which types of information the SNMP manager needs (if your application is using an SNMP manager). Choose the particular MIB objects that you want through the management software.
- Decide how many trap hosts you need. In some network configurations, you may want to have a primary trap host with one other workstation also receiving traps for redundancy. In a distributed or segmented network, you may want to have more trap hosts enabled. You can configure up to five trap hosts per SNMP agent; that is, one agent can report to a maximum of five hosts.
- Designate a management station or stations. The CSS is an agent in the SNMP network scheme. The agent is already embedded in the CSS when you boot up the device. All you need to do is configure the SNMP parameters on the CSS.

Defining the CSS as an SNMP Agent

This section describes how to define the CSS as an SNMP agent. It includes the following topics:

- SNMP Agent Quick Start
- Configuring an SNMP Community
- Configuring an SNMP Contact
- Configuring an SNMP Location
- Configuring an SNMP Name
- Configuring SNMP Generic Traps
- Configuring an SNMP Trap-Host
- Configuring SNMP Source Traps
- Configuring SNMP Auth-Traps
- Configuring SNMP Enterprise Traps
- Configuring SNMP Reload-Enable

SNMP Agent Quick Start

Table 12-1 provides a quick overview of the steps required to configure the CSS as an SNMP agent. 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, see the sections following Table 12-1.

Table 12-1 Quick Start for Defining the CSS as an SNMP Agent

Task and Command Example

1. Define the SNMP community strings for each access type, read-only (for a GET action) or read-write (for a GET and SET action). This step is required for using SNMP on the CSS.
`(config)# snmp community public read-only`
`(config)# snmp community private read-write`

2. (Optional) Provide the SNMP contact name. `(config)# snmp contact "fred n mandy"`

Table 12-1 Quick Start for Defining the CSS as an SNMP Agent (continued)

Task and Command Example
3. (Optional) Provide an SNMP contact location. `(config)# snmp location "Operations"`
4. (Optional) Provide the SNMP device name. `(config)# snmp name "arrowpoint.com"`
5. (Optional) Turn on generic traps. `(config)# snmp trap-type generic`

6. Assign trap receivers and SNMP community (required if configuring SNMP traps). You can specify a maximum of five trap hosts. By default, all traps are disabled.The **snmp trap-host** IP address corresponds to the SNMP host configured to receive traps. The community information provided at the end of the **trap-host** command is included in the trap and can be used by the management station to filter incoming traps.
`(config)# snmp trap-host 172.16.3.6 trap`
`(config)# snmp trap-host 172.16.8.4 trap`

7. (Optional) Turn on authentication failure traps. An authentication failure occurs if an unauthorized SNMP manager sends an invalid or incorrect community name to an SNMP agent. If an authentication failure occurs, the agent sends an authentication trap to the trap host (or hosts depending on how many trap hosts are configured). `(config)# snmp auth-traps`

8. (Optional) Enable global enterprise traps.
`(config)# snmp trap-type enterprise`
Enable a specific enterprise trap type. For example, you can set a trap to notify the trap host of failed login attempts. Login failure traps provide the username and source IP address of the person who failed to log in.
`(config)# snmp trap-type enterprise login-failure`

Table 12-1 Quick Start for Defining the CSS as an SNMP Agent (continued)

Task and Command Example
9. (Optional) Configure the trap host for reload enable ability. Reload enable allows a management station with the proper WRITE community privilege to reboot the CSS. `(config)# snmp reload-enable 100`
10. (Optional) Configure special enterprise trap thresholds to notify the trap host of Denial of Service (DoS) attacks on your system. For example, you can set a trap threshold to notify the trap host of DoS attacks with illegal addresses, either source or destination. `(config)# snmp trap-type enterprise dos-illegal-attack trap-threshold 1`

Configuring an SNMP Community

Use the **snmp community** command to set or modify SNMP community names and access privileges. You may specify as many community names as you wish.

⚠ **Caution** You must define the community strings for each access type (read-only or read-write) before you use SNMP on the CSS. The CSS is inaccessible until you specify a read community string.

The syntax for this global configuration mode command is:

snmp community *community_name* [**read-only**|**read-write**]

The variables and options for this command are:

- *community_name* - The SNMP community name for this system. Enter an unquoted text string with no space and a maximum of 12 characters.
- **read-only** - Allows read-only access for this community.
- **read-write** - Allows read-write access for this community.

For example:

```
(config)# snmp community sqa read-write
```

To remove a community name, enter:

```
(config)# no snmp community sqa
```

Configuring an SNMP Contact

Use the **snmp contact** command to set or modify the contact name for the SNMP system. You can specify only one contact name. The syntax for this global configuration mode command is:

snmp contact *"contact_name"*

Enter the contact name as a quoted text string with a maximum of 255 characters including spaces. You can also include information on how to contact the person; for example, a phone number or e-mail address.

For example:

```
(config)# snmp contact "Fred N. Mandy"
```

To remove the specified SNMP contact name and reset it to the default of "Cisco Systems, Content Network Systems", enter:

```
(config)# no snmp contact
```

Configuring an SNMP Location

Use the **snmp location** command to set or modify the SNMP system location. You can specify only one location. The syntax for this global configuration mode command is:

snmp location *"location"*

Enter the location as the physical location of the system. Enter a quoted text string with a maximum of 255 characters.

For example:

```
(config)# snmp location "sqa_lab1"
```

To remove the specified SNMP system location and reset it to the default of "Customer Premises", enter:

```
(config)# no snmp location
```

Configuring an SNMP Name

Use the **snmp name** command to set or modify the SNMP name for this system. You can specify only one name. The syntax for this global configuration mode command is:

snmp name *"name"*

Enter the SNMP name as the unique name assigned to a system by the administrator. Enter a quoted text string with a maximum of 255 characters. The standard name convention is the system's fully qualified domain name (for example, sqa@arrowpoint.com).

For example:

```
(config)# snmp name "sqa@arrowpoint.com"
```

To remove the SNMP name for a system and reset it to the default of "Support", enter:

```
(config)# no snmp name
```

Configuring SNMP Generic Traps

Use the **snmp trap-type generic** command to enable SNMP generic trap types. The generic SNMP traps consist of cold start, warm start, link down, and link up.

Note For details on SNMP traps (and associated MIB objects) loaded as part of the CSS software, see the "CSS SNMP Traps" section.

For example:

```
(config)# snmp trap-type generic
```

To disable a generic trap, enter:

```
(config)# no snmp trap-type generic
```

Configuring an SNMP Trap-Host

Use the **snmp trap-host** command to set or modify the SNMP host to receive traps from a CSS. You can specify a maximum of five hosts. The syntax for this global configuration mode command is:

snmp trap-host *ip_or_host community_name*

The variables for this command are:

- *ip_or_host* - The IP address or host name of an SNMP host that has been configured to receive traps. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1) or in mnemonic host-name format (for example, myhost.mydomain.com).
- *community_name* - The community name to use when sending traps to the specified SNMP host. Enter an unquoted text string with no spaces and a maximum of 12 characters.

For example:

```
(config)# snmp trap-host 172.16.3.6 sqalab1
```

To remove a specified trap host, enter:

```
(config)# no snmp trap-host 172.16.3.6
```

Configuring SNMP Source Traps

Use the **snmp trap-source** command to set the source IP address in the traps generated by the CSS. The syntax of this global configuration mode command is:

snmp trap-source [**egress-port**|**specified** *source_ip_address*]

The options and variable for this command are:

- **egress-port** - Obtains the source IP address for the SNMP traps from the VLAN circuit IP address configured on the egress port used to send the trap. You do not need to enter an IP address because the address is determined dynamically by the CSS.
- **specified** *source_ip_address* - Allows you to enter the IP address to be used in the source IP field of the traps. Enter the IP address in dotted-decimal notation (for example, 192.168.11.1)

For example:

```
(config)# snmp trap-source egress-port
```

To return SNMP source traps to the default of the management port IP address, enter:

```
(config)# no snmp trap-source
```

Configuring SNMP Auth-Traps

Use the **snmp auth-traps** command to enable reception of SNMP authentication traps. The CSS generates these traps when an SNMP management station attempts to access your system with invalid community names.

Note For details on SNMP traps (and associated MIB objects) loaded as part of the CSS software, see the "CSS SNMP Traps" section.

For example:

```
(config)# snmp auth-traps
```

To disable reception of authentication traps, enter:

```
(config)# no snmp auth-traps
```

Configuring SNMP Enterprise Traps

Use the **snmp trap-type enterprise** command to enable SNMP enterprise trap types. You can enable the CSS to generate enterprise traps when:

- Denial of Service attack events occur
- A login fails
- A CSS service transitions state
- A power supply transitions state
- A module is inserted into a powered-on CSS chassis
- An Inter-Switch Communications (ISC) LifeTick failure message occurs

Use the **no** form of the **snmp trap-type enterprise** command to prevent the CSS from generating a trap when a specific condition occurs.

For details on SNMP traps (and associated MIB objects) loaded as part of the CSS software, see the "CSS SNMP Traps" section. For information on configuring Denial of Service enterprise traps, see the "Configuring Denial of Service (DoS)" section.

The syntax for this global configuration mode command is:

snmp trap-type enterprise {*dos_attack_type* {**trap-threshold** *threshold_value*}|**chmgr-module-transition**|**chmgr-ps-transition** |**isc-lifetick-failure**|**login-failure**|**reload**|**redundancy-transition** |**service-transition**}

The options for this command are as follows:

- **snmp trap-type enterprise** - Enables enterprise traps. You must enable enterprise traps before you configure an enterprise trap option.

- *dos_attack_type* - (Optional) Generates SNMP enterprise traps when a Denial of Service (DoS) attack event occurs. One trap is generated each second when the number of attacks during that second exceeds the threshold for the configured DoS attack type. See the "Configuring Denial of Service (DoS)" section for details.
- **trap-threshold** *threshold_value* - (Optional) Overrides a default trap threshold. For the *threshold_value*, enter a number from 1 to 65535. See the "Configuring Denial of Service (DoS)" section for details.
- **chmgr-module-transition** - Generates SNMP enterprise traps if a module (for example, SCM or SSL) is inserted into or removed from a powered-on CSS 11503 or CSS 11506.
- **chmgr-ps-transition** - Generates SNMP enterprise traps when the CSS 11503 or CSS 11506 power supply changes state (powered off or on, or removed from the CSS).
- **isc-lifetick-failure** - Generates SNMP enterprise traps when an ISC LifeTick failure message occurs on a CSS. A LifeTick message occurs four times a second between ports in an Adaptive Session Redundancy (ASR) configuration. If a port does not receive a LifeTick message within one second from its corresponding port due to a software or hardware failure, an ISC LifeTick failure message occurs.
- **login-failure** - Generates SNMP enterprise traps when a CSS login failure occurs. The CSS also generates an alert-level log message.
- **reload** - Generates SNMP enterprise traps when a CSS reboot occurs. The CSS also generates a trap when a reboot is initiated directly through SNMP.
- **redundancy-transition** - Generates SNMP enterprise traps when the CSS redundancy transitions state.
- **service-transition** - Generates SNMP enterprise traps when a CSS service transitions state. A trap is generated when a service fails and when a failed service resumes proper operation.

To enable an SNMP enterprise trap when a CSS login failure occurs, enter:

```
(config)# snmp trap-type enterprise login-failure
```

To disable all enterprise traps, enter:

```
(config)# no snmp trap-type enterprise
```

To disable a specific enabled enterprise trap, use the **no** form of the **snmp trap-type enterprise** command. To prevent the CSS from generating traps when a power supply fails, enter:

```
(config)# no snmp trap-type enterprise chmgr-ps-transition
```

Configuring SNMP Reload-Enable

Use the **snmp reload-enable** command to reboot the CSS using SNMP. The syntax and options for this global configuration mode command are:

- **snmp reload-enable** - Allows any SNMP write to the apSnmpExtReloadSet object to force a CSS reboot. The reload object, apSnmpExtReloadSet, is located at 1.3.6.1.4.1.2467.1.22.7. You can find this object in the CSS Enterprise MIB, snmpext.mib.
- **snmp reload-enable** *reload_value* - Allows an SNMP write equal to the *reload_value* to force a CSS reboot.

Enter the *reload_value* as the object used to control apSnmpExtReloadSet, providing the SNMP-based reboot. When the object is set to 0, an SNMP reboot is not allowed. When the object is set from 1 to 232, a reboot may be caused with any write value to apSnmpExtReloadSet. For security purposes, this object always returns 0 when read.

For example:

```
(config)# snmp reload-enable
```

To prevent users from rebooting the CSS using SNMP (default behavior), enter:

```
(config)# no snmp reload-enable
```

Configuring Denial of Service (DoS)

You can configure special enterprise traps to notify the trap host of Denial of Service (DoS) attacks on your system. You can also use the CLI to display detailed information about DoS attacks and reset the DoS statistics for your CSS to zero.

Ensure you first enable SNMP enterprise traps using the **snmp trap-type enterprise** command before you configure the CSS to generate SNMP enterprise traps when a DoS attack event occurs. For information, see the "Configuring SNMP Enterprise Traps" section.

This section includes the following topics:

- DoS Quick Start
- Defining a DoS SNMP Trap-Type
- Displaying DoS Configurations

DoS Quick Start

Table 12-2 provides a quick overview of the steps required to configure the CSS as an SNMP agent. 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, see the sections following Table 12-2.

Table 12-2 Denial of Service Configuration Quick Start

Task and Command Example
1. Set the trap threshold to notify the trap host of DoS attacks with illegal addresses, either source or destination. `(config)# snmp trap-type enterprise dos-illegal-attack trap-threshold 1`
2. Set the trap threshold to notify the trap host of DoS LAND attacks. `(config)# snmp trap-type enterprise dos-land-attack trap-threshold 1`
3. Set the trap threshold to notify the trap host of DoS smurf attacks. `(config)# snmp trap-type enterprise dos-smurf-attack trap-threshold 1`

Table 12-2 Denial of Service Configuration Quick Start (continued)

Task and Command Example
4. Set the trap threshold to notify the trap host of DoS SYN attacks. `(config)# snmp trap-type enterprise dos-syn-attack trap-threshold 10`

5. Display information about DoS attacks.
`(config)# show dos summary`
`(config)# show dos`

6. Reset the DoS statistics for a CSS to zero, as required. `(config)# zero dos statistics`

Defining a DoS SNMP Trap-Type

Use the **snmp trap-type enterprise** command to enable the CSS to generate SNMP enterprise traps when a DoS attack event occurs. One trap is generated each second when the number of attacks during that second exceeds the threshold for the configured DoS attack type. For details on SNMP traps (and associated MIB objects) loaded as part of the CSS software, see the "CSS SNMP Traps" section.

Ensure you first enable SNMP enterprise traps using the **snmp trap-type enterprise** command before you configure the CSS to generate SNMP enterprise traps when a DoS attack event occurs. For information, see the "Configuring SNMP Enterprise Traps" section.

The syntax for this global configuration mode command is:

snmp trap-type enterprise *dos_attack_type* {**trap-threshold** *threshold_value*}

The *dos_attack_type* variable is the type of DoS attack event to trap. The options include:

- **dos-illegal-attack** - Generates traps for illegal addresses, either source or destination. Illegal addresses are loopback source addresses, broadcast source addresses, loopback destination addresses, multicast source addresses, or source addresses that you own. The default trap threshold for this type of attack is 1 per second.
- **dos-land-attack** - Generates traps for packets that have identical source and destination addresses. The default trap threshold for this type of attack is 1 per second.
- **dos-smurf-attack** - Generates traps when the number of pings with a broadcast destination address exceeds the threshold value. The default trap threshold for this type of attack is 1 per second.
- **dos-syn-attack** - Generates traps when the number of TCP connections that are initiated by a source, but not followed with an acknowledgment (ACK) frame to complete the 3-way TCP handshake, exceeds the threshold value. The default trap threshold for this type of attack is 10 per second.

Use the **trap-threshold** option to override a default trap threshold. For the *threshold_value*, enter a number from 1 to 65535.

For example, to enable the CSS to generate traps for packets that have identical source and destination addresses, enter:

```
(config)# snmp trap-type enterprise dos-land-attack
```

To prevent the CSS from generating DoS attack event traps, enter:

```
(config)# no snmp trap-type enterprise dos_attack_type
```

Displaying DoS Configurations

Use the **show dos** command to display detailed information about DoS attacks on each CSS Session Processor. The **show dos** command displays the following information:

- The total number of attacks since booting the CSS
- The types of attacks and the maximum number of these attacks per second
- The first and last occurrence of an attack
- The source and destination IP addresses

A CSS can display a maximum of 50 of the most recent attack events for each SP. For example:

- A CSS 11501 with one SP can display a maximum of 50 events.
- A CSS 11503 with a maximum of three SPs can display a maximum of 150 events.
- A CSS 11506 with a maximum of six SPs can display a maximum of 300 events.

If multiple attacks occur with the same DoS type and source and destination address, an attempt is made to merge them as one event. This merging of events reduces the number of displayed events.

Use the **show dos summary** command to display a summary of information about DoS attacks.

For example:

```
(config)# show dos summary
```

Table 12-3 describes the fields in the **show dos** command output.

Table 12-3 Field Descriptions for the show dos Command

Field	Description
Total Attacks	The total number of DoS attacks detected since the CSS was booted. The type of attacks that are listed along with their number of occurrences are: • **SYN Attacks** - TCP connections that are initiated by a source but are not followed with an ACK frame to complete the three way TCP handshake • **LAND Attacks** - Packets that have identical source and destination addresses • **Zero Port Attacks** - Frames that contain source or destination TCP or UDP ports equal to zero **Note** Older SmartBits software may send frames containing source or destination ports equal to zero. The CSS logs them as DoS attacks and drops these frames. • **Illegal Src Attacks** - Illegal source addresses • **Illegal Dst Attacks** - Illegal destination addresses • **Smurf Attacks** - Pings with a broadcast destination address
First Attack Detected	The first time a DoS attack was detected.
Last Attack Detected	The last time a DoS attack was detected.

Table 12-3 Field Descriptions for the show dos Command (continued)

Field	Description
Maximum per second	The maximum number of events per second. Use the maximum events per second information to set SNMP trap threshold values. The maximum number of events per second is the maximum for each SP. • For a CSS 11506, which may have up to six SPs, the maximum rate per second may be as high as six times the value appearing in this field. • For a CSS 11503, which may have up to three SPs, the maximum rate per second may be as high as three times the value appearing in this field.
DoS Attack Event	Details for each detected attack event, up to a maximum of 50 events per SP.
First Attack	The first time the attack event occurred.
Last Attack	The last time the attack event occurred.
Source/Destination Address	The source and destination addresses for the attack event.
Event Type	The type of event.
Total Attacks	The total number of attack occurrences for the event.

Displaying the SNMP Configuration

After you configure SNMP, display the SNMP configuration. For example:

```
(config)# show running-config global
```

Refer to Chapter 3, Managing the CSS Software for details on the **show running-config** command and its output.

Managing SNMP on the CSS

This section describes the activities that you need to perform to manage SNMP on the CSS. This section includes the following topics:

- Enabling SNMP Manager Access to the CSS
- Using the CSS to Look Up MIB Objects
- Reading Logs
- Setting RMON Alarms

Enabling SNMP Manager Access to the CSS

By default, the CSS enables SNMP access to its command base. You must first create community strings using the **snmp community** command before you can use SNMP in the CSS. See the "Configuring an SNMP Community" section for details.

Note SNMP is not a secure network environment. Do not use SNMP by itself to provide security for your network.

Using the CSS to Look Up MIB Objects

To look up a MIB object, including the variables that make up the object, perform the following steps:

1. Access global configuration mode by entering:

```
# config
```

2. Access rmon-alarm mode by entering:

```
(config)# rmon-alarm index_number
```

   where *index_number* is the RMON alarm index. The RMON alarm index identifies the alarm to the CSS. Refer to Chapter 13, Configuring Remote Monitoring (RMON) for information on RMON.

3. Display the MIB object by entering:

```
(config-rmonalarm[1])# lookup object
```

   where *object* is the name of the MIB object.

   You can look up a specific object, or you can use the question mark (?) character as a wildcard to help you complete your request.

For example, suppose you want to look up a MIB object but you are not sure of its exact name. You already know that the MIB you want is part of the apFlowMgrExt group of objects. In this case, specify the **lookup** command with the question mark (**?**) character, as shown.

```
(config-rmonalarm[1])# lookup apFlowMgrExt?

apFlowMgrExtDoSAttackEventType
apFlowMgrExtDoSAttackEventCount
apFlowMgrExtDoSAttackIndex
apFlowMgrExtDosTotalSmurfAttacks
apFlowMgrExtDosTotalIllegalSourceAttacks
apFlowMgrExtDosTotalZeroPortAttacks
apFlowMgrExtDosTotalLandAttacks
apFlowMgrExtDosTotalSynAttacks
apFlowMgrExtDosTotalAttacks
apFlowMgrExtIdleTimer
apFlowMgrExtPortIdleValue
apFlowMgrExtPortIdle
apFlowMgrExtReserveCleanTimer
apFlowMgrExtPermanentPort4
apFlowMgrExtPermanentPort3
```

```
apFlowMgrExtPermanentPort2
apFlowMgrExtPermanentPort1
apFlowMgrExtFlowTraceDuration
apFlowMgrExtFlowTraceMaxFileSize
apFlowMgrExtFlowTraceState
```

The previous example shows that using the question mark (?) character as a wildcard returns information about the apFlowMgrExt MIB object. You can also enter the **lookup** command on the exact MIB you want and view its description without using the question mark (?) character. For example:

```
(config-rmonalarm[1])# lookup apFlowMgrExtDOSAttackEventCount

ASN Name:            apFlowMgrExtDOSAttackEventCount
MIB:                 flowmgrext
Object Identifer:    1.3.6.1.4.1.2467.1.36.27.1.6
Argument Type:       Integer
Range:               0-4294967295
Description:
   This is the number of times this DoS attack had occurred.
```

You can also display a list of all the Enterprise MIBs by using the **lookup** command without any MIB object names, as shown in the following example:

```
(config-rmonalarm[1])# lookup ?
```

The **lookup** command omits MIB objects of type *string* and *MAC address*.

Useful MIB Information

Table 12-4 lists some of the MIB groups that provide useful information about the CSS.

Table 12-4 CSS MIB Information

MIB Name	Description
RFC 1398	Ethernet statistics
RFC 1493	Bridge information
RFC 1757	RMON statistics
svcExt.mib	Service variables (including TCP connections)
cntExt.mib	Content rule variables (including frame statistics)

Table 12-4 CSS MIB Information (continued)

MIB Name	Description
ownExt.mib	Owner statistics (including frame and bytes counts)
cntsvcExt.mib	Services per content rule statistics (including frames, bytes, hits)
chassis MgrExt	Provides useful information about the CSS chassis and it allows you to correlate the slot number and port number to the ifIndex number

Reading Logs

The traplog file contains all of the traps, both generic and enterprise, that have occurred. The network device writes to the traplog file about whether or not the SNMP trap configuration is enabled.

Use the **show log** command to show the trap log since the last CSS reboot. For example:

```
# show log traplog
```

By default, the following events generate level critical-2 messages:

- Link Up
- Link Down
- Cold Start
- Warm Start
- Service Down
- Service Suspended

All other SNMP traps generate level notice-5 messages by default.

Setting RMON Alarms

An RMON alarm allows you to monitor a MIB object for a desired transitory state. Refer to Chapter 13, Configuring Remote Monitoring (RMON) for information about commands available in the RMON alarm mode.

CSS SNMP Traps

Table 12-5 lists the SNMP traps supported by the CSS.

Table 12-5 SNMP Traps

Name/MIB	Enterprise Object ID (OID)	Generic	Specific	Parameters
coldStart	<sysObjectID>	0	0	———————————
warmStart	<sysObjectID	1	0	———————————
linkDown	<sysObjectID>	2	0	ifIndex 1.3.6.1.2.1.2.2.1.1
linkUp	<sysObjectID>	3	0	ifIndex 1.3.6.1.2.1.2.2.1.1
authenticationFailure	<sysObjectID>	4	0	———————————
egpNeighborLoss	<sysObjectID>	5	0	———————————
apFlowMgrExtDosSynTrap (flowMgrExt.mib)	1.3.6.1.4.1.2467.1.36	6	1	apFlowMgrExtDOSAttackEventString 1.3.6.1.4.1.2467.1.36.28.1.8 apFlowMgrExtDOSAttackEventIntervalCount 1.3.6.1.4.1.2467.1.36.28.1.9 apFlowMgrExtDOSAttackEventCount 1.3.6.1.4.1.2467.1.36.28.1.6
apFlowMgrExtDosLandTrap (flowMgrExt.mib)	1.3.6.1.4.1.2467.1.36	6	2	apFlowMgrExtDOSAttackEventString 1.3.6.1.4.1.2467.1.36.28.1.8 apFlowMgrExtDOSAttackEventIntervalCount 1.3.6.1.4.1.2467.1.36.28.1.9 apFlowMgrExtDOSAttackEventCount 1.3.6.1.4.1.2467.1.36.28.1.6
apFlowMgrExtDosIllegalTrap (flowMgrExt.mib)	1.3.6.1.4.1.2467.1.36	6	3	apFlowMgrExtDOSAttackEventString 1.3.6.1.4.1.2467.1.36.28.1.8 apFlowMgrExtDOSAttackEventIntervalCount 1.3.6.1.4.1.2467.1.36.28.1.9 apFlowMgrExtDOSAttackEventCount 1.3.6.1.4.1.2467.1.36.28.1.6

Table 12-5 SNMP Traps (continued)

Name/MIB	Enterprise Object ID (OID)	Generic	Specific	Parameters
apFlowMgrExtDosSmurfTrap (flowMgrExt.mib)	1.3.6.1.4.1.2467.1.36	6	5	apFlowMgrExtDOSAttackEventString 1.3.6.1.4.1.2467.1.36.28.1.8 apFlowMgrExtDOSAttackEventInterval Count 1.3.6.1.4.1.2467.1.36.28.1.9 apFlowMgrExtDOSAttackEventCount 1.3.6.1.4.1.2467.1.36.28.1.6
apIpv4RedundancyTrap (apIpv4.mib)	1.3.6.1.4.1.2467.1.9.1	6	1	apIpv4TrapEventText 1.3.6.1.4.1.2467.1.9.34.0 apIpv4RedundancyState 1.3.6.1.4.1.2467.1.9.19.0 apIpv4RedundancyIf 1.3.6.1.4.1.2467.1.9.20.0 apIpv4RedundancyMaster 1.3.6.1.4.1.2467.1.9.21.0
apSnmpExtReloadTrap (snmpExt.mib)	1.3.6.1.4.1.2467.1.22	6	1	apSnmpExtTrapEventText 1.3.6.1.4.1.2467.1.22.27.0
apSvcTransitionTrap (svcExt.mib)	1.3.6.1.4.1.2467.1.15	6	1	apSvcTrapEventText 1.3.6.1.4.1.2467.1.15.10.0
apTermSessLoginFailureTrap (terminalMgmt.mib)	1.3.6.1.4.1.2467.1.11	6	1	apTermSessLoginFailureInfo 1.3.6.1.4.1.2467.1.11.3.0
apChassisMgrExtPsTrap (chassisMgrExt.mib)	1.3.6.1.4.1.2467.1.34	6	1	apChassisMgrExtTrapPsEventText 1.3.6.1.4.1.2467.1.34.24.0
apChassisMgrModuleTrap (chassisMgrExt.mib)	1.3.6.1.4.1.2467.1.34	6	2	apChassisMgrExtTrapModuleEventText 1.3.6.1.4.1.2467.1.34.25.0
apEnetISCLifetickTrap (enetExt.mib)	1.3.6.1.4.1.2467.1.39	6	1	apEnetISCLifetickEventText 1.3.6.1.4.1.2467.1.39.8.0

CSS MIBs

Table 12-6 describes the CSS MIB objects directly under the CSS Enterprise MIB (Object Identifier 1.3.6.1.4.1.2467). The MIBs listed in this table are a representation of the CSS content-specific MIB objects. To find out how you can look up object information, see the "Using the CSS to Look Up MIB Objects" section.

Table 12-6 MIB Branches Under the CSS Enterprise MIB

MIB Filename	MIB Module Description	Related CLI Commands
aclExt.mib (OID 1.3.6.1.4.1.2467.1.23)	The CSS access control list (ACL) clause table	`(config-acl)# ?`

ap64Stats.mib
(OID 1.3.6.1.4.1.2467.1.44)
	The 64-bit statistical aggregation of RMON (RFC1757), MIB-II (RFC 1213) and EtherErrors (RFC 1398)	`# show rmon ?`
		`# show mibii ?`
		`# show ether-errors ?`

apent.mib (OID 1.3.6.1.4.1.2467.1)	CSS Enterprise MIB branch hierarchy	——————————
apIpv4.mib (OID 1.3.6.1.4.1.2467.1.9.1)	MIB support for IPv4 global information, box-to-box redundancy	`(config)# ip ?`
apIpv4Arp.mib (OID 1.3.6.1.4.1.2467.1.9.4)	MIB support for IPv4 ARP	`(config)# arp ?`
apIpv4Dns.mib (OID 1.3.6.1.4.1.2467.1.9.7)	MIB support for IPv4 DNS resolver configuration	`(config)# dns ?`
apIpv4Host.mib (OID 1.3.6.1.4.1.2467.1.9.6)	MIB support for IPv4 host table	`(config)# host ?`
apIpv4Interface.mib (OID 1.3.6.1.4.1.2467.1.9.2)	MIB support for IPv4 interfaces, box-to-box redundancy	`(config-ip)# ?`
apIpv4Ospf.mib (OID 1.3.6.1.4.1.2467.1.9.3.2)	MIB support for the Open Shortest Path First (OSPF) protocol	`(config)# ospf ?`
apIpv4Redundancy.mib (OID 1.3.6.1.4.1.2467.1.9.8)	MIB support for IPv4 redundancy	`(config-ip)# redundancy ?`

Table 12-6 MIB Branches Under the CSS Enterprise MIB (continued)

MIB Filename	MIB Module Description	Related CLI Commands
apIpv4Rip.mib (OID 1.3.6.1.4.1.2467.1.9.3.1)	MIB support for the Routing Information Protocol (RIP)	`(config-ip)#` **`rip ?`**
apIpv4Sntp.mib (OID 1.3.6.1.4.1.2467.1.9.9)	MIB support for the Simple Network Time Protocol (SNTP)	`(config)#` **`sntp ?`**
apIpv4StaticRoutes.mib (OID 1.3.6.1.4.1.2467.1.9.5)	MIB support for IPv4 static routes	`(config)#` **`ip route ?`**
appExt.mib (OID 1.3.6.1.4.1.2467.1.32)	MIB support for Application Peering Protocol (APP) configurations	`(config)#` **`app ?`**
boomClientExt.mib (OID 1.3.6.1.4.1.2467.1.62)	Configuration and monitoring of Content Routing Agent (CRA) parameters	`(config)#` **`dns-boomerang client ?`**
bootExt.mib (OID 1.3.6.1.4.1.2467.1.31)	MIB support for system boot adminstration	`(config-boot)#` **`?`**
bridgeExt.mib (OID 1.3.6.1.4.1.2467.1.14)	Configuration and monitoring of bridge-related parameters	`(config)#` **`bridge ?`**
cappUdpExt.mib (OID 1.3.6.1.4.1.2467.1.52)	Application Peering Protocol-User Datagram Protocol (APP-UDP) global statistical information and security configuration settings	`(config)#` **`app-udp ?`**
cctExt.mib (OID 1.3.6.1.4.1.2467.1.29)	CSS circuit information, box-to-box redundancy	`(config)#` **`circuit ?`**
chassisMgrExt.mib (OID 1.3.6.1.4.1.2467.1.34)	MIB for the CSS chassis manager	`#` **`show chassis ?`**
cntdnsExt.mib (OID 1.3.6.1.4.1.2467.1.41)	Content rule Domain Name Service (DNS) statistics	`(config)#` **`dns hotlist ?`**
cntExt.mib (OID 1.3.6.1.4.1.2467.1.16)	Content rule table	`(config-owner-content)#` **`?`**
cnthotExt.mib (OID 1.3.6.1.4.1.2467.1.35)	Content rule hot list	`(config-owner-content)#` **`hotlist ?`**

Table 12-6 MIB Branches Under the CSS Enterprise MIB (continued)

MIB Filename	MIB Module Description	Related CLI Commands

cntsvcExt.mib
(OID 1.3.6.1.4.1.2467.1.18)
	Monitoring of services attached to content rules	`(config-owner-content)#` **`add service ?`**
		`(config-owner-content)#` **`remove service ?`**

csaExt.mib (OID 1.3.6.1.4.1.2467.1.59)	Configuration and monitoring of Client Side Accelerator (CSA) parameters on a CSS	`(config)#` **`dns-server ?`**
dfpExt.mib (OID 1.3.6.1.4.1.2467.1.65)	MIB support for Dynamic Feedback Protocol (DFP) statistics and configuration	`(config)#` **`dfp ?`**
dnshotExt.mib (OID 1.3.6.1.4.1.2467.1.48)	DNS hot list	`(config)#` **`domain hotlist ?`**
dnsServerExt.mib (OID 1.3.6.1.4.1.2467.1.40)	MIB support for DNS server	`(config)#` **`dns-server ?`**
domainCacheExt.mib (OID 1.3.6.1.4.1.2467.1.60)	Configuration management for the domain cache on the CSA in the CSS	`(config)#` **`dns-server domain-cache ?`**
dqlExt.mib (OID 1.3.6.1.4.1.2467.1.51)	Domain Qualifier Lists (DQLs)	`(config-dql [name])#` **`?`**
enetExt.mib (OID 1.3.6.1.4.1.2467.1.39)	Configuration of the PHY state for Ethernet ports	`(config-interface)#` **`phy ?`**
eqlExt.mib (OID 1.3.6.1.4.1.2467.1.42)	Extension Qualifier Lists (EQLs)	`(config-eql [name])#`
fileExt.mib (OID 1.3.6.1.4.1.2467.1.61)	File extensions to support network management movement to/from the CSS, and to examine and modify the existing file structure	——————————
flowMgrExt.mib (OID 1.3.6.1.4.1.2467.1.36)	MIB for the flow manager module	`(config)#` **`flow ?`**
ftpExt.mib (OID 1.3.6.1.4.1.2467.1.30)	MIB support for File Transfer Protocol (FTP) transfer administration records	`(config)#` **`ftp-record ?`**

Table 12-6 MIB Branches Under the CSS Enterprise MIB (continued)

MIB Filename	MIB Module Description	Related CLI Commands
grpExt.mib (OID 1.3.6.1.4.1.2467.1.17)	Configuration of all group-related parameters	`(config-group)#` **`?`**

grpsvcExt.mib
(OID 1.3.6.1.4.1.2467.1.19)
	Groups attached to services	`(config-group)#` **`add service ?`**
		`(config-group)#` **`remove service ?`**

httpExt.mib (OID 1.3.6.1.4.1.2467.1.47)	MIB support for HTTP transfer administration records	——————————
kalExt.mib (OID 1.3.6.1.4.1.2467.1.46)	Configuration of keepalive mode	`(config-keepalive)#` **`?`**
logExt.mib (OID 1.3.6.1.4.1.2467.1.20)	CSS logging functionality	`(config)#` **`logging ?`**
nqlExt.mib (OID 1.3.6.1.4.1.2467.1.50)	Describes the CSS network qualifier lists (NQLs)	`(config-nql [name])#` **`?`**
ownExt.mib (OID 1.3.6.1.4.1.2467.1.25)	Web host owner information	`(config-owner)#` **`?`**
plucExt.mib (OID 1.3.6.1.4.1.2467.1.56)	Proximity Lookup Client functionality	`(config)#` **`proximity cache ?`**
probeRttExt.mib (OID 1.3.6.1.4.1.2467.1.55)	Tiered Proximity Service RTT Probe Module functionality	`(config)#` **`proximity probe rtt ?`**
proxDbExt.mib (OID 1.3.6.1.4.1.2467.1.54)	Tiered Proximity Database (PDB) functionality; contains all configuration, statistic, and metric objects	`(config)#` **`proximity db ?`**
publishExt.mib (OID 1.3.6.1.4.1.2467.1.57)	Publisher and subscriber services	`(config-service)#` **`publisher ?`**
qosExt.mib (OID 1.3.6.1.4.1.2467.1.28)	CSS MIB module quality of service (QoS) class definitions (the QoS class of this known piece of content)	——————————
radiusClientExt.mib (OID 1.3.6.1.4.1.2467.1.12)	CSS extensions to the client side of the Remote Access Dial-in User Service (RADIUS) authentication protocol	`(config)#` **`radius-server ?`**

Table 12-6 MIB Branches Under the CSS Enterprise MIB (continued)

MIB Filename	MIB Module Description	Related CLI Commands
schedExt.mib (OID 1.3.6.1.4.1.2467.1.45)	MIB support for CLI command scheduler records	`(config)# cmd-scheduler ?`
securityMgrExt.mib (OID 1.3.6.1.4.1.2467.1.13)	CSS MIB objects for the network security manager	`(config)# username ?`
snmpExt.mib (OID 1.3.6.1.4.1.2467.1.22)	SNMP traps and communities	`(config)# snmp ?`
sshdExt.mib (OID 1.3.6.1.4.1.2467.1.43)	MIB support for the Secure Shell Daemon server (SSHD)	`(config)# sshd ?`

sslExt.mib
(OID 1.3.6.1.4.1.2467.1.63)
	MIB support for Secure Sockets Layer (SSL) file associations for SSL certificates and keys for the SSL Acceleration Module	`(config)# ssl cert ?`
		`(config)# ssl rsakey ?`
		`(config)# ssl dakey ?`
		`(config)# ssl dhparm ?`

ssllExt.mib (OID 1.3.6.1.4.1.2467.1.64)	MIB support for SSL proxy list elements and cipher suite objects for the SSL Acceleration Module	`(ssl-proxy-list[name])# element ?`
subscribeExt.mib (OID 1.3.6.1.4.1.2467.1.58)	CSS Enterprise subscriber	`(config-service)# subscriber ?`
svcExt.mib (OID 1.3.6.1.4.1.2467.1.15)	Configuration and monitoring of all service-related parameters	`(config-service)# ?`
tacacsExt.mib (OID 1.3.6.1.4.1.2467.1.66)	CSS extensions to the client side of the Terminal Access Controller Access Control System (TACACS+) authentication protocol	`(config)# tacacs-server ?`
tagExt.mib (OID 1.3.6.1.4.1.2467.1.53)	Content tag lists	`(config)# header-field-group ?`

terminalMgmt.mib
(OID 1.3.6.1.4.1.2467.1.11)
	MIB support for terminal options	`# terminal ?`
		`# restrict ?`

urqlExt.mib (OID 1.3.6.1.4.1.2467.1.49)	Uniform resource locator qualifier lists (URQL)	`(config-urql [name])# ?`

Where to Go Next

Chapter 13, Configuring Remote Monitoring (RMON), describes how to describes how to configure RMON on the CSS.

Configuring Remote Monitoring (RMON)

This chapter provides information on configuring the Remote Monitoring (RMON) features of your CSS. Information in this chapter applies to all CSS models, except where noted.

This chapter contains the following major sections:

- RMON Overview
- RMON Configuration Considerations
- Configuring an RMON Event
- Configuring an RMON Alarm
- Configuring an RMON History
- Viewing RMON Information
- RMON Configuration in a Startup-Config File

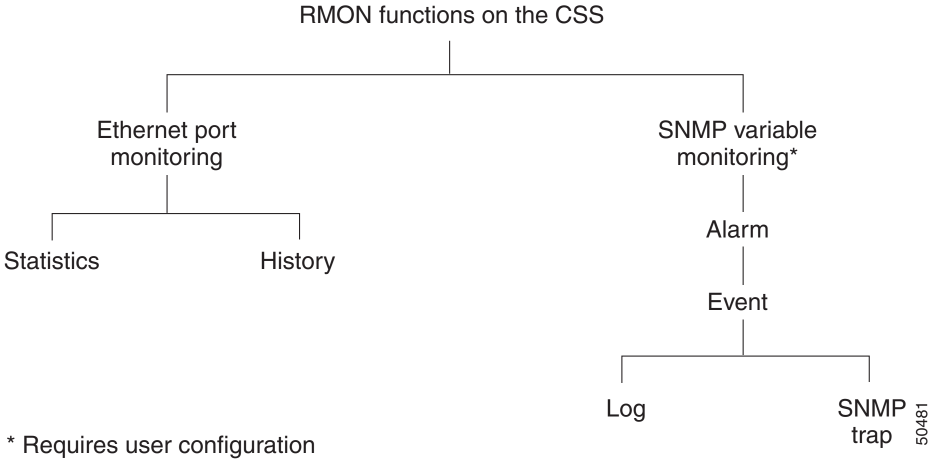
RMON Overview

RMON allows you to remotely monitor and analyze the activity of packets on CSS Ethernet ports. RMON also allows alarm configuration for monitoring MIB objects, and the event configuration to notify you of these alarm conditions. For detailed information about RMON and its MIB objects, refer to RFC 1757.

The version of RMON provided on the CSS is a subset of the RMON-1 groups (Figure 13-1). The CSS supports the following groups:

- Group 1 - (Statistics) Provides data about all Ethernet ports on a CSS. You cannot configure RMON statistics. You can only view them.
- Group 2 - (History) Provides data about the Ethernet ports over an historical period. Histories are preconfigured for each port. You can configure additional port histories.
- Group 3 - (Alarm) Allows you to create an alarm and configure the conditions, based on a MIB object, to trigger an alarm when changes are detected.
- Group 9 - (Event) Allows you to create an event and configure the event action that is to be performed when an associated alarm occurs.

Figure 13-1 Supported RMON Functions on the CSS


RMON functions on the CSS
Ethernet port monitoring
SNMP variable monitoring*
Statistics
History
Alarm
Event
Log
SNMP trap
50481
* Requires user configuration

RMON Configuration Considerations

Consider the following points before you implement RMON functionality on your CSS:

- You can configure an RMON event, alarm, and history. You cannot configure CSS attributes for RMON statistics. Statistics for the ports are viewable only by using the **show rmon** command.
- You cannot change the configuration for an RMON history after you activate it. If you need to change the RMON history configuration after activation, you must delete it first and then recreate the RMON history with the necessary changes. You can change your RMON history configuration at any time before you activate it.
- You must assign an RMON event to an RMON alarm before the alarm can be activated. The event must exist and must be activated before it can be assigned to an RMON alarm.
- RMON histories are preconfigured for each Ethernet port. Though these histories cannot be deleted or modified, you can add history entries for a port. For more information on the preconfigured histories and adding more history entries, see the "Configuring an RMON History" section.

Configuring an RMON Event

An RMON event is the action that occurs when an associated RMON alarm is triggered. When an alarm event occurs, it can be configured to generate a log event, a trap to an SNMP network management station, or both. For information on viewing alarm events in log files, see the "Viewing Events in a Log File" section. Refer to Chapter 12, Configuring Simple Network Management Protocol (SNMP) for information on configuring SNMP on your CSS.

The following sections describe how to configure an RMON event.

- RMON Event Configuration Quick Start
- Creating an Index for an RMON Event
- Deleting an RMON Event Index
- Setting the RMON Event Attributes
- Activating an RMON Event
- Suspending an RMON Alarm

RMON Event Configuration Quick Start

Table 13-1 provides a quick overview of the steps required to configure the attributes for an RMON event. 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, see the sections following Table 13-1.

Table 13-1 RMON Event Configuration Quick Start

Steps and Possible Settings
1. Create an RMON event index from global configuration mode. Enter an integer from 1 to 65534. `(config)# rmon-event 1`
2. Assign an existing SNMP community for this event. The specified *community_name* is the name of an SNMP community configured using the **snmp trap-host** command. This step is required only if the traps are sent to an SNMP network management station. `(config-rmonevent[1])# community moonbase_alpha`

Table 13-1 RMON Event Configuration Quick Start (continued)

Steps and Possible Settings
3. Provide a description for the event. Enter a quoted text string with a maximum of 127 characters including spaces. `(config-rmonevent[1])# description “This event occurs when service connections exceed 100”`
4. Assign the owner who defined and is using the resources of the event. Enter a quoted string with a maximum of 127 characters including spaces. You must define the owner before you can activate the event. `(config-rmonevent[1])# owner “Boston Tech Lab”`
5. Specify the type of event notification. The type determines where the notification is sent. `(config-rmonevent[1])# type log-and-trap`
6. Activate the event. `(config-rmonevent[1])# active`

For information on configuring an alarm and associating this event to an alarm, see the “Configuring an RMON Alarm” section.

Creating an Index for an RMON Event

The RMON event index identifies the event to the CSS. This index allows you to assign specific configuration attributes to the RMON event. When you create an RMON event index, you access the configuration mode for that event automatically.

Use the **rmon-event** *index* command to create an event index. You can access this command from any configuration mode except the boot and RMON alarm configuration modes. The *index* is a number from 1 to 65534.

Note The RMON event index 65535 is administratively predefined and cannot be modified. If you enter this index number, a message similar to the following appears: `%% Index internally used. Administrative control not allowed.`

For example, to create an RMON event with an identifier of 1, access global configuration mode and enter:

```
(config)# rmon-event 1
```

To view a list of existing RMON event configuration identifiers, enter:

```
(config)# rmon-event ?
```

After you create the index for the event, the prompt changes to (config-rmonevent[1]). Define the event as described in the "Setting the RMON Event Attributes" section.

Modifying the Attributes for an Existing RMON Event Index

Use the **suspend** command to deactivate the RMON event and make attribute changes.

Deleting an RMON Event Index

If you have an active RMON event index that you no longer need, use the **no rmon-event** command. This command is available in the RMON alarm, RMON event, RMON history, and global configuration modes.

To delete RMON event 1 and its configuration, enter:

```
(config)# no rmon-event 1
Delete Event <1>,[y/n]:y
```

Setting the RMON Event Attributes

After you create an RMON event index, access RMON event configuration mode for the event identifier and set its attributes. This section includes the following topics:

- Defining an Event Community
- Describing an Event
- Assigning an Owner
- Defining the Notification of an Event

After you set the attributes, activate the event as described in the "Activating an RMON Event" section.

If an RMON event is activated and you want to make modifications to certain event attributes, you must first suspend the RMON event (as described in the "Suspending an RMON Event" section). Ensure the RMON event is not assigned to an RMON alarm.

Defining an Event Community

When an alarm event occurs and the event is configured to send an SNMP trap, the CSS sends the trap to the trap host with the specified community. If no community is specified the CSS automatically uses the default event community of "public".

Use the **community** *community_name* command to define a community to an unactivated event. The *community_name* variable is the name of the SNMP community you configured using the **snmp trap-host** command (refer to Chapter 12, Configuring Simple Network Management Protocol (SNMP)).

To view a list of currently configured community strings (configured using the **snmp trap-host** command), enter **rmon-event community ?**

For example, to define the SNMP *moonbase_alpha* community for this event, enter:

```
(config-rmonevent[1])# community moonbase_alpha
```

To reset the community back to public, enter:

```
(config-rmonevent[1])# no community
```

Describing an Event

When an alarm event occurs, the CSS sends a description with the event notification. Because a description is not generated automatically, you must provide one. Use the **description** *"description"* command to provide a description. The *description* variable is the description for the RMON event. Enter a quoted text string with a maximum of 127 characters.

To provide a description for the event, enter:

```
(config-rmonevent[1])# description "This event occurs when service
connections exceed 100"
```

To remove the description from the event, enter:

```
(config-rmonevent[1])# no description
```

Assigning an Owner

You must define the entity who configured this RMON event and is using the resources assigned to it. Use the **owner** "*owner_name*" command to define the owner. The *owner_name* variable is a quoted text string with a maximum of 127 characters. The owner for the event must be the same as the owner for the alarm.

To define the owner named Boston Tech Lab, enter:

```
(config-rmonevent[1])# owner "Boston Tech Lab"
```

To remove the owner of the RMON event, enter:

```
(config-rmonevent[1])# no owner
```

You must reassign an owner before you can reactivate the RMON event.

Defining the Notification of an Event

When an RMON event occurs, the event type determines where the CSS sends the event notification.

- A log event type designates that the event notification is made in a CSS log location (for example, CSS disk log file or session). For information on viewing log files, see the "Viewing Events in a Log File" section.

  To define the event as a log type (default), enter:

  ```
  (config-rmonevent[1])# type log
  ```

- A trap event type designates that a trap is sent to a SNMP network management station. To define the event as a trap type, enter:

  ```
  (config-rmonevent[1])# type trap
  ```

Note When you want the event to send a trap to a network management station, you need to configure SNMP. Refer to Chapter 12, Configuring Simple Network Management Protocol (SNMP) for information on SNMP.

- You can also designate that the event type is both log and trap. To define the event as both log and trap types, enter:

```
(config-rmonevent[1])# type log-and-trap
```

To reset the RMON event type back to log, enter:

```
(config-rmonevent[1])# no type
```

Activating an RMON Event

After you configure the event attributes, activate the event. However, before you can activate an event, you must specify the owner of the event as described in the "Assigning an Owner" section.

To activate the event, enter:

```
(config-rmonevent[1])# active
```

When activating an RMON event, once an RMON event is activated and you want to make modifications to certain event attributes, you must first suspend the RMON event. Ensure the RMON event is not assigned to an RMON alarm.

Suspending an RMON Event

Suspending an RMON event deactivates it, allowing you to make changes to its configuration settings. Use the **suspend** command to suspend an event.

When you suspend an RMON event, ensure that the event is not assigned to an RMON alarm.

For example:

```
(config-rmonevent[1])# suspend
```

Configuring an RMON Alarm

An RMON alarm allows you to monitor a MIB object for a desired transitory state. An alarm periodically takes samples of the object's value and compares them to the configured thresholds.

RMON allows you to configure two types of sampling, absolute and delta:

- Absolute sampling compares the sample value directly to the threshold. This sampling is similar to a gauge, recording values that go up or down.
- Delta sampling subtracts the current sample value from the last sample taken and then compares the difference to the threshold. This sampling is similar to a counter, recording a value that is constantly increasing.

When a sample value crosses an alarm threshold, an associated event is generated. To limit the number of generated events, only one event is generated when a threshold is crossed. The CSS does not generate additional events until an opposite threshold is crossed. For example, when a rising threshold is crossed, one event is generated. The next event occurs only when a falling threshold is crossed.

When you associate an event to an alarm and an alarm occurs, the event defines the action the CSS takes when an alarm occurs. For more information on events, see the "Configuring an RMON Event" section.

Figure 13-2 is an example of absolute sampling.

Figure 13-2 Example of Absolute Sampling

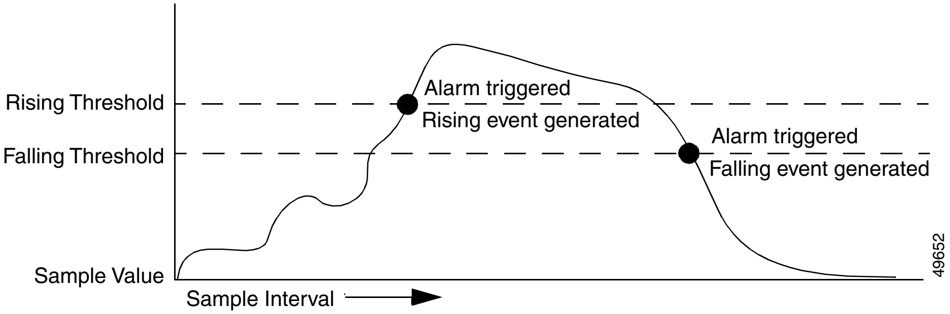
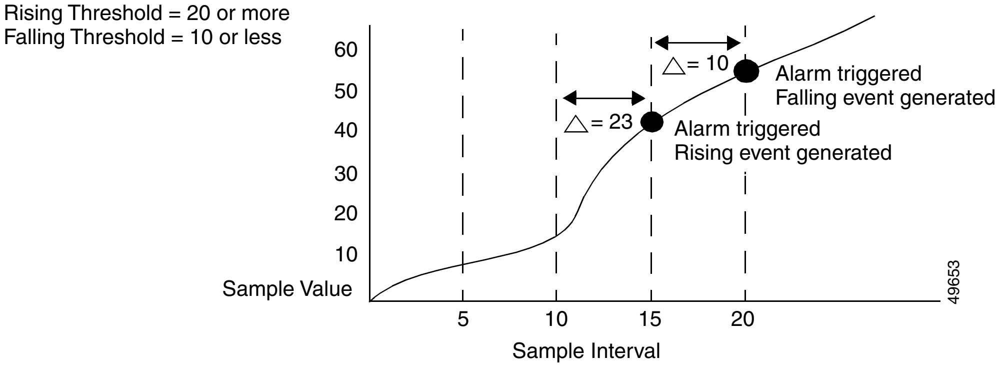


Figure 13-3 is an example of delta sampling.

Figure 13-3 Example of Delta Sampling



This section includes the following topics:

- RMON Alarm Configuration Quick Start
- Creating an Index for an RMON Alarm
- Deleting an RMON Alarm Index
- Setting the RMON Alarm Attributes
- Activating an RMON Alarm
- Suspending an RMON Alarm

RMON Alarm Configuration Quick Start

Table 13-2 provides a quick overview of the steps required to configure the attributes for an RMON alarm. 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, see the sections following Table 13-2.

Table 13-2 RMON Alarm Configuration Quick Start

Steps and Possible Settings
1. Create an RMON alarm index from global configuration mode. Enter an integer from 1 to 65534. `(config)# rmon-alarm 1`
2. Assign the owner who defined and is using the resources of the alarm. Enter a quoted string with a maximum of 127 characters including spaces. The owner must be the same as the owner for the event. `(config-rmonalarm[1])# owner “Boston Tech Lab”`
3. Define the MIB object for the sample variable. For example, for the current number of connections for this service, enter **apSvcConnections**. To see a list of objects, use the **sample-variable ?** command. For detailed information about an object, use the **lookup** command. `(config-rmonalarm[1])# sample-variable apSvcConnections`
4. Define the sampling type. The options are **absolute** or **delta** (default). `(config-rmonalarm[1])# sample-type absolute`

Table 13-2 RMON Alarm Configuration Quick Start (continued)

Steps and Possible Settings
5. Define the startup alarm type. The options are **falling**, **rising** (default), or **rising-and-falling**. `(config-rmonalarm[1])# startup-type rising-and-falling`
6. Define the rising threshold. Enter an integer from 0 (default) to 4294967295. `(config-rmonalarm[1])# rising-threshold 100`
7. Associate the rising event with an existing RMON event. Enter an integer from 0 to 65534. If you enter 0, no event is generated. `(config-rmonalarm[1])# rising-event 1`
8. Define the falling threshold. Enter an integer from 0 (default) to 4294967295. `(config-rmonalarm[1])# falling-threshold 90`
9. Associate the falling event with an existing RMON event. Enter a number from 0 to 65534. If you enter 0, no event is generated. `(config-rmonalarm[1])# falling-event 2`
10. Specify the sampling interval for the RMON alarm. The interval is in seconds. Enter an integer from 1 to 65535 (default is 300). `(config-rmonalarm[1])# sample-interval 30`
11. Activate the alarm. `(config-rmonalarm[1])# active`

Creating an Index for an RMON Alarm

The RMON alarm index identifies the alarm to the CSS. This index allows you to assign specific configuration attributes to the RMON alarm. When you create an RMON alarm index, you access the configuration mode for that alarm automatically.

Use the **rmon-alarm** *index* command to create an alarm index. You can access this command from any configuration mode except the boot and RMON history configuration modes. The *index* is an integer from 1 to 65534.

Note The RMON alarm index 65535 is administratively predefined and cannot be modified. If you enter this index number, a message similar to the following appears: `%% Index internally used. Administrative control not allowed.`

To create an RMON alarm with an identifier of 1, access global configuration mode and enter:

```
(config)# rmon-alarm 1
```

To see a list of existing RMON alarm configuration identifiers, enter **rmon-alarm ?**.

After you create the identifier for the alarm, the prompt changes to (`config-rmonalarm[1]`). Now you can define the alarm as described in the "Setting the RMON Alarm Attributes" section.

Modifying the Attributes for an Existing RMON Alarm Index

Use the **suspend** command to deactivate the RMON alarm and make attribute changes.

Deleting an RMON Alarm Index

Use the **no rmon-alarm** command if you have an active RMON alarm index that you no longer need and want to delete it. This command is available in the RMON alarm, RMON event, RMON history, and global configuration modes.

For example, to delete RMON alarm 1 and its configuration, enter:

```
(config)# no rmon-alarm 1
Delete Alarm <1>,[y/n]:y
```

Setting the RMON Alarm Attributes

After you create an RMON alarm index, access RMON alarm configuration mode for an existing inactive alarm identifier and set its attributes. This sections includes the following topics:

- Assigning an Owner
- Finding and Defining a Sample Variable
- Defining the Absolute or Delta Sampling Method
- Defining a Rising Threshold and Rising Event
- Defining a Falling Threshold and Index
- Defining a Startup Alarm
- Defining the Sampling Interval

After you set all of the attributes, activate the alarm as described in the "Activating an RMON Alarm" section.

If an RMON alarm is activated and you want to make modifications to certain alarm attributes, you must first suspend the RMON alarm (as described in the "Suspending an RMON Alarm" section).

Assigning an Owner

Define the owner who configured the RMON alarm and is using the resources assigned to the alarm. To define the owner, use the **owner** "*owner_name*" command. You must reassign an owner before you can reactivate the RMON alarm.

The *owner_name* variable is a quoted text string with a maximum of 127 characters. Enter the same name as the owner of the event.

To define the owner named Boston Tech Lab, enter:

```
(config-rmonalarm[1])# owner "Boston Tech Lab"
```

To remove the owner of the RMON alarm, enter:

```
(config-rmonalarm[1])# no owner
```

Finding and Defining a Sample Variable

For an alarm condition, RMON samples a configured sample variable associated with a MIB object. MIB objects to consider include:

- svcExt.mib - Contains service objects (for example, apSvcConnections is the MIB object for the current number of TCP connections to this service).
- cntExt.mib - Contains content rule objects (for example, apCntHits is the MIB object for the total number of hits on this service for this content rule).

Refer to Chapter 12, Configuring Simple Network Management Protocol (SNMP) for information on CSS Enterprise MIBs.

Use the **lookup** command to look up a MIB object and view its description. For example, to view the description for the apSvcConnections object, enter:

```
(config-rmonalarm[1])# lookup apSvcConnections
ASN Name:            apSvcConnections
MIB:                 svcext
Object Identifier:   1.3.6.1.4.1.2467.1.15.2.1.20
Argument Type:       Integer
Range:               0-4294967295
Description:
   The current number of TCP connections to this service
```

Use the **sample-variable** *mib_object* command to specify the sample variable for this RMON alarm. For example, to define the apSvcConnections MIB object for the current number of service connections, enter:

```
(config-rmonalarm[1])# sample-variable apSvcConnections
```

To see a list of SNMP variables, use the **sample-variable ?** command. For example:

```
(config-rmonalarm[1])# sample-variable ?

               apSvcLoadInfoTimeout
               apSvcLoadSvcStatRptTimeout
               apSvcLoadEnable
               apSvcLoadDecayInterval
               apSvcLoadStepStatic
               apSvcLoadStepSize
               apSvcLoadThreshold
               ...
```

To remove the sample variable, enter:

```
(config-rmonalarm[1])# no sample-variable
```

Defining the Absolute or Delta Sampling Method

When you configure an alarm, you can define the sampling method to compare the sample value of a MIB object to either:

- The configured threshold directly. This sampling is similar to a gauge, recording the value as it fluctuates up and down (see Figure 13-2).
- The previous sampling, and then their difference is compared to the configured threshold. This sampling is similar to a counter, recording the value that constantly increases (see Figure 13-3).

Absolute sampling compares the sample value to the configured threshold. For example, if you want to know when 30,000 service connections occur on the CSS during a sample interval, configure the apSvcConnections MIB object with absolute sampling. The apSvcConnections object is the current number of connections on a service. To define an absolute sampling, enter:

```
(config-rmonalarm[1])# sample-type absolute
```

Delta sampling (the default sampling method) compares the current sample value with the previous sample and compares their difference to the configured threshold. For example, if you want to know when the number of content rule hits increase by 100,000 as compared to its previous sampling, configure the apCntHits MIB object with delta sampling. apCntHits is an ever-increasing count of hits.

To define a delta sampling, enter:

```
(config-rmonalarm[1])# sample-type delta
```

To reset the sample type to delta sampling, enter:

```
(config-rmonalarm[1])# no sample-type
```

Defining a Rising Threshold and Rising Event

If you want to be notified when a sampling is greater than or equal to a specific number, set a rising threshold and associate it with a configured RMON event.

You must create an RMON event before you can associate it with an alarm.

For a single rising alarm event to occur, a sampled value is greater than or equal to the rising threshold value, and the value at the last sampling interval is less than this threshold.

- Use the **rising-threshold** *rising_value* command to set the threshold for the alarm. The *rising_value* variable is the threshold for the rising sample type. Enter an integer from 0 (default) to 4294967295.

  To set the rising threshold value of 100, enter:

  ```
  (config-rmonalarm[1])# rising-threshold 100
  ```

  To reset the rising threshold to 0, enter:

  ```
  (config-rmonalarm[1])# no rising-threshold
  ```

- Use the **rising-event** *rising_index* command to associate a configured event to the RMON alarm when the sampled value exceeds the rising threshold value. The *rising_index* variable is the event index used when a rising threshold is crossed. Enter a previously created RMON event index (see the "Creating an Index for an RMON Event" section). If you enter 0, no event is generated.

  To associate the threshold to RMON event 1, enter:

  ```
  (config-rmonalarm[1])# rising-event 1
  ```

  To see a list of RMON events, enter:

  ```
  (config-rmonalarm[1])# rising-event ?
  ```

  To reset the rising event to 0 (no event is generated), enter:

  ```
  (config-rmonalarm[1])# no rising-event
  ```

Defining a Falling Threshold and Index

If you want to be notified when a sampling is less than or equal to a specific number, set a falling threshold and associate it to a configured event.

Note You must create an RMON event before you can associate it with an alarm.

For a single falling alarm event to occur, a sampled value is less than or equal to the falling threshold value, and the value at the last sampling interval is greater than this threshold.

- Use the **falling-threshold** *falling_value* command to set the threshold for the alarm. The *falling_value* variable is the threshold for the falling sample type. Enter an integer from 0 (default) to 4294967295.

  To set the falling threshold value of 90, enter:

  ```
  (config-rmonalarm[1])# falling-threshold 90
  ```

  To reset the falling threshold to 0, enter:

  ```
  (config-rmonalarm[1])# no falling-threshold
  ```

- Use the **falling-event** *falling_index* command to associate a configured event to the RMON alarm when the sampled value exceeds the falling threshold value. The *falling_index* variable is the event index used when a falling threshold is crossed. Enter a previously created RMON event index (see the "Creating an Index for an RMON Event" section). If you enter 0, no event is generated.

  To associate the threshold to RMON event 2, enter:

  ```
  (config-rmonalarm[1])# falling-event 2
  ```

  To see a list of RMON events, enter:

  ```
  (config-rmonalarm[1])# falling-event ?
  ```

  To reset the falling event to 0, enter:

  ```
  (config-rmonalarm[1])# no falling-event
  ```

Defining a Startup Alarm

A startup alarm allows the CSS to generate an alarm when the first sample triggers a falling threshold or rising threshold (default).

- A startup falling alarm occurs when the first sample is less than or equal to the falling threshold. To enable this alarm, enter:

  ```
  (config-rmonalarm[1])# startup-type falling
  ```

- A startup rising alarm occurs when the first sample is greater than or equal to the rising threshold. To enable this alarm, enter:

  ```
  (config-rmonalarm[1])# startup-type rising
  ```

- To enable an alarm when either a falling or rising threshold is triggered, enter:

  ```
  (config-rmonalarm[1])# startup-type rising-and-falling
  ```

To reset the startup alarm to a rising threshold alarm, enter:

```
(config-rmonalarm[1])# no startup-type
```

Defining the Sampling Interval

The sampling interval is the time interval, in seconds, over which the data is sampled and compared with the rising and falling thresholds. Use the **sample-interval** *interval* command to specify the sampling interval for the RMON alarm. The *interval* variable is the number of seconds from 1 to 65535. The default is 300 seconds.

To enter a sampling interval of 60 seconds, enter:

```
(config-rmonalarm[1])# sample-interval 60
```

With delta sampling, set the sampling interval short enough so the sampled variable, which has a tendency to go up and down very fast, does not wrap during a single sampling period.

To reset the sample interval to 300, enter:

```
(config-rmonalarm[1])# no sample-interval
```

Activating an RMON Alarm

After you configure the alarm attributes, you can activate the alarm. However, before you can activate an alarm, you must specify all attributes for the alarm.

To activate the alarm, enter:

```
(config-rmonalarm[1])# active
```

When activating an RMON alarm, once an RMON alarm is activated and you want to make modifications to certain alarm attributes, you must first suspend the RMON alarm.

Suspending an RMON Alarm

Suspending an RMON alarm deactivates it, allowing you to make changes to its configuration settings. To suspend an alarm, use the **suspend** command.

For example:

```
(config-rmonalarm[1])# suspend
```

Configuring an RMON History

You can configure the operation of the RMON history that periodically samples any CSS Ethernet port for statistical data. All ports are preconfigured with histories for 30-second and 30-minute intervals, and 50 buckets with one sample for each bucket. However, you can create additional histories for a specific port. The creation of an RMON history for a port allows you to configure the time interval to take the sample and the number of samples you want to save.

You can view the statistical information for the history by using the **show rmon-history** command. For more information about viewing the history, see the "Viewing History" section.

This section includes the following topics:

- RMON History Configuration Quick Start
- Creating an Index for an RMON History
- Deleting an RMON History Index
- Setting the RMON History Attributes
- Activating an RMON History Entry

RMON History Configuration Quick Start

Table 13-3 provides a quick overview of the steps required to configure the attributes for an RMON history. 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, see the sections following Table 13-3.

Table 13-3 RMON History Configuration Quick Start

Steps and Possible Settings
1. Create an RMON history from global configuration mode. Enter an integer from 1 to 65535. `(config)# rmon-history 5`
2. Assign the owner who defined and is using the history resources. Enter a maximum of 32 characters. `(config-rmonhistory[5])# owner "Boston_Tech_Lab"`
3. Define the data source object for the Ethernet port. The port is identified by an ifIndex data object identifier. `(config-rmonhistory[5])# data-source ifIndex.3`
4. Define the time interval for the history. Enter a number from 1 to 3600 seconds. The default is 1800 seconds. `(config-rmonhistory[5])# interval 60`

Table 13-3 RMON History Configuration Quick Start (continued)

Steps and Possible Settings
5. Define the bucket count for the interval. Enter an integer from 1 to 65535. The default is 50. `(config-rmonhistory[5])# requested-buckets 25`
6. Activate the history. `(config-rmonhistory[5])# active`

Creating an Index for an RMON History

The RMON history index identifies the history to the CSS. The RMON history index allows you to assign specific configuration attributes to the RMON history index. When you create an RMON history index, you access the configuration mode for that history automatically.

To create a history index, use the **rmon-history** *index* command from any configuration mode except boot configuration mode. The *index* is an integer from 1 to 65534.

Note The RMON history index 65535 is administratively predefined and cannot be modified. If you enter this index number, a message similar to the following appears: `%% Index internally used. Administrative control not allowed.`

To create an RMON history identifier 5, access global configuration mode and enter:

```
(config)# rmon-history 5
```

To see a list of existing RMON history configuration identifiers, enter **rmon-history ?**.

After you create the identifier for the history, the prompt changes to (`config-rmonhistory[1]`). Now you can define the history as described in the "Setting the RMON History Attributes" section.

Modifying the Attributes for an Existing RMON History Index

If create an RMON history index but have not activated it, you can modify the attributes of the history index (as described in the "Setting the RMON History Attributes" section). Once the RMON history index is activated, you cannot modify its attributes. You must delete the history index (see the "Deleting an RMON History Index" section), recreate it, and respecify the alarm index attributes.

Deleting an RMON History Index

If you have an active RMON history index that requires changes to its attributes or you no longer need it, delete the RMON history index. Before you delete a history index that requires changes, note the settings for its attributes.

To delete an RMON history configuration identifier, use the **no rmon-history** command. This command is available in the RMON alarm, RMON event, RMON history, and global configuration modes.

For example, to delete RMON history 5 and its configuration, enter:

```
(config)# no rmon-history 5
Delete History <5>,[y/n]:y
```

After you delete the history identifier to change its attributes, recreate it as described in the "Creating an Index for an RMON History" section.

Setting the RMON History Attributes

After you create an RMON history or access RMON history configuration mode for an existing inactive alarm, set the RMON history attributes. This section includes the following topics:

- Defining the Data Object
- Assigning an Owner
- Defining the Bucket Count
- Defining the Bucket Interval

After you set the attributes, activate the history as described in the "Activating an RMON History Entry" section.

Defining the Data Object

After you create a history, you must associate it with a CSS Fast Ethernet or Gigabit Ethernet port. To define the data object, use the **data-source** *port* command. The *port* is identified by an ifIndex data object identifier. For example, if your CSS has 12 Ethernet ports, they have data object IDs of ifIndex.1 through ifIndex.12. The Ethernet management port has an ID of ifIndex.14.

To define Ethernet port 4, enter:

```
(config-rmonhistory[5])# data-source ifIndex.4
```

To see a list of data object IDs for all of the CSS Ethernet ports, enter:

```
(config-rmonhistory[5])# show interface
```

Assigning an Owner

Define the owner who configured the RMON history and is using the resources assigned to it. Use the **owner** *owner_name* command to define the owner. The *owner_name* variable is a quoted text string with a maximum of 32 characters.

For example, to define an owner named Boston Tech Lab, enter:

```
(config-rmonhistory[5])# owner “Boston Tech Lab”
```

Defining the Bucket Count

You can define a bucket count, which is the number of discrete sampling intervals over which data is saved for a history entry. Use the **requested-buckets** *count* command to define a bucket count. The *count* variable is an integer from 1 to 65535. The default is 50.

To define a bucket count of 25, enter:

```
(config-rmonhistory[5])# requested-buckets 25
```

Defining the Bucket Interval

You can specify the time interval, in seconds, to take a bucket sample for an RMON history operation. Use the **interval** *value* command to set this interval. Enter an integer from 1 to 3600 seconds. The default is 1800 (30 minutes).

To define a time interval of 60 seconds, enter:

```
(config-rmonhistory[5])# interval 60
```

Activating an RMON History Entry

After you configure the history attributes, you can activate the history for the port. Use the **active** command to activate an RMON history entry. Before you activate the history, make sure you finish configuring it and are satisfied with the RMON history settings. After you activate a history, you cannot modify its configuration settings. The only way to change the history is to delete it, and then recreate it.

Before activating this command, you must specify the owner of the RMON history entry.

To activate the history, enter:

```
(config-rmonhistory[5])# active
```

Viewing RMON Information

RMON information includes:

- Ethernet port statistics and history data that you can view from the CSS through **show** commands.
- Alarm event notifications that are sent to log locations on the CSS or an SNMP network management station. Refer to Chapter 12, Configuring Simple Network Management Protocol (SNMP) for information on configuring SNMP on the CSS.

The following sections provide information on:

- Viewing Statistics
- Viewing History
- Viewing Events in a Log File

Viewing Statistics

RMON statistics provide a summary of data received over the Fast Ethernet or Gigabit Ethernet ports. You can view RMON statistics either in a CSS CLI session through the **show rmon** command, the **show ether-errors** command (see Chapter 5, Configuring Interfaces and Circuits), or directly through an SNMP network management station by using ether-stats MIB objects (refer to RFC 1398).

The CSS **show rmon** command allows you to display the extended 64-bit RMON statistics for a specific Ethernet port or all Ethernet ports in the CSS. The CSS Enterprise ap64Stats MIB defines these statistics. You can also display the RFC 1757 32-bit statistics by adding the **-32** suffix to the **show rmon** command.

To display the RMON statistics for all ports in the CSS, enter:

```
# show rmon
```

To display the RFC 1757 32-bit statistics, enter:

```
# show rmon-32
```

To display the RMON statistics for a specified port in the CSS, enter:

```
# show rmon port_name
```

The *port_name* variable is the name of the physical port (for example, ethernet-4). Enter the *port_name* variable as a case-sensitive, unquoted text string.

To display the RFC 1757 32-bit statistics, enter **show rmon-32** p*ort_name*.

To see a list of ports, enter:

```
# show rmon ?
```

To display the extended RMON statistics for the Ethernet-4 port in the CSS, enter:

```
# show rmon ethernet-4
```

Table 13-4 lists and describes the fields in the **show rmon** command output.

Table 13-4 Field Descriptions for the show rmon Command

Field	Description
Bytes	The total number of received bytes.
Packets	The total number of received packets (including bad packets, broadcast packets, and multicast packets).
Broadcast Packets	The total number of good received packets that were directed to the broadcast address. The number of broadcast packets does not include multicast packets.
Multicast Packets	The total number of good received packets that were directed to a multicast address. The number of multicast packets does not include packets directed to the broadcast address.
CRC Alignment Errors	The total number of packets received that had a length (excluding framing bits, but including FCS octets) from 64 and 1518 octets, but had an FCS Error, a bad Frame Check Sequence (FCS) with an integral number of octets, or an Alignment Error, a bad FCS with a non-integral number of octets.
Oversize Packets	The total number of received packets that were longer than 1518 octets (excluding framing bits, but including FCS octets) and were otherwise well formed.

Table 13-4 Field Descriptions for the show rmon Command (continued)

Field	Description
Undersize Packets	The total number of received packets that were less than 64 octets long (excluding framing bits, but including FCS octets) and were otherwise well-formed.
Fragments	The total number of received packets that were less than 64 octets in length (excluding framing bits but including FCS octets) and had an FCS Error, a bad Frame Check Sequence (FCS) with an integral number of octets, or an Alignment Error, a bad FCS with a non-integral number of octets. It is normal for fragment statistics to increment because the CSS counts both runts (which are normal occurrences due to collisions) and noise hits.
Drop Events	The total number of events in which packets were dropped by the probe due to lack of resources. This number is not necessarily the number of packets dropped; it is the number of times this condition has been detected.
Slobbers	An internal counter. This field is always zero.
Jabbers	The total number of packets received that were longer than 1518 octets (excluding framing bits, but including FCS octets), and had an FCS Error, a bad Frame Check Sequence (FCS) with an integral number of octets, or Alignment Error, a bad FCS with a non-integral number of octets. This definition of jabber is different than the definition in IEEE-802.3, section 8.2.1.5, 10BASE5, and section 10.3.1.4, 10BASE2. These documents define jabber as the condition where any packet exceeds 20 ms. The allowed range to detect jabber is between 20 and 150 ms.

Table 13-4 Field Descriptions for the show rmon Command (continued)

Field	Description
Collisions	The best estimate of the total number of collisions on this Ethernet segment. The returned value depends on the location of the RMON probe. Section 8.2.1.3, 10BASE-5, and section 10.3.1.3, 10BASE-2, of IEEE standard 802.3 states that a station must detect a collision, in the receive mode, if three or more stations are transmitting simultaneously. A repeater port must detect a collision when two or more stations are transmitting simultaneously. Thus, a probe placed on a repeater port might record more collisions than would a probe connected to a station on the same segment. Probe location plays a much smaller role when considering 10BASE-T. IEEE standard 802.3 14.2.1.4, 10BASE-T, defines a collision as the simultaneous presence of signals on the DO and RD circuits (transmitting and receiving at the same time). A 10BASE-T station can detect collisions only when it is transmitting. Probes placed on a station and a repeater should report the same number of collisions. Ideally, an RMON probe inside a repeater should report collisions between the repeater and one or more other hosts (transmit collisions as defined by IEEE 802.3k), plus receiver collisions observed on any coaxial segments to which the repeater is connected.
Packets (0-64) Packets (65-127) Packets (128-255) Packets (256-511) Packets (512-1023) Packets (1024-1518)	The total number of received packets (including bad packets) that were between the following octets in length inclusive (excluding framing bits but including FCS octets): • 0 to 64 • 65 to 127 • 128 to 255 • 256 to 511 • 512 to 1023 • 1024 to 1518

Clearing RMON Statistics

Use the **clear statistics** *port_name* command to reset the RMON statistics on a CSS Ethernet port to zero. The *port_name* variable is the name of the physical port (for example, ethernet-4). Enter the *port_name* variabl as a case-sensitive, unquoted text string.

To clear the statistics for Ethernet port 1, enter:

```
# clear statistics Ethernet-1
```

To see a list of ports, enter:

```
# clear statistics ?
```

Note When you reset RMON statistics on a CSS Ethernet port to zero, the Ethernet errors and MIB-II statistics for the port are also reset to zero.

Viewing History

You can display the default and configured RMON history information for a specific Ethernet port or all Ethernet ports in the CSS. For information on configuring an RMON history, see the "Configuring an RMON History" section.

By default, the CSS maintains two tables of history statistics for each port. One table contains the last 50 samples at 30-second intervals. The other table contains 50 samples at 30-minute intervals. You cannot modify the configuration for these histories.

- To view the RMON history for all ports in the CSS, enter:

  ```
  # show rmon-history
  ```

- To display the RMON history for a specified port, enter:

  ```
  # show rmon-history port_name
  ```

  To see a list of ports in the CSS, enter:

  ```
  # show rmon-history ?
  ```

- To display the RMON history for a specified port and history index, enter:

```
# show rmon-history port_name history_index
```

To view the history 5 for the Ethernet-4 port, enter:

```
# show rmon-history ethernet-4 5
```

To see a list of history indexes associated with a specified port, enter:

```
# show rmon-history port_name ?
```

To see a list of histories for the Ethernet-4 port, enter:

```
# show rmon-history ethernet-4 ?
```

Table 13-5 lists and describes the fields in the **show rmon-history** command output.

Table 13-5 Field Descriptions for the show rmon-history Command

Field	Description
Owner	The owner who configured the entry and is using the resources assigned to it.
Start Time	The time when the bucket sampling started.
Interval	The time interval, in seconds, when RMON takes a bucket sample.
Buckets	The number of discrete sampling intervals over which data is saved for the history.
Time	The time that the sample was taken.
Sample	The number of the sample.

Table 13-5 Field Descriptions for the show rmon-history Command (continued)

Field	Description
Octets	The total number of octets of data (including those in bad packets) received on the network, excluding framing bits but including FCS octets. You can use this object as a reasonable estimate of Ethernet utilization. If greater precision is desired, sample the Ethernet statistic packet and octet objects before and after a common interval. The differences in the sampled values are packets (Pkts) and Octets, respectively, and the number of seconds in the Interval. These values are used to calculate the utilization of a 10 Mbps Ethernet port as follows: $\text{Utilization} = \dfrac{\text{Pkts} * (9.6 + 6.4) + (\text{Octets} * .8)}{\text{Interval} * 10{,}000}$ The result of this equation is the utilization value, which is the utilization percentage of the Ethernet segment on a scale of 0 to 100 percent.
Packets	The total number of received packets (including bad packets, broadcast packets, and multicast packets).
Errors	The total number of errors that RMON received for this port.
Util%	The bandwidth utilization percentage of the Ethernet segment on a scale of 0 to 100 percent.

Viewing Events in a Log File

The CSS can send notifications of RMON alarm events to a traplog file or a configured log location, such as a log file on the CSS disk, a CSS session, a host syslog daemon, or an e-mail address. The notification itself displays the time that the event occurred, the event number, and its configured description in parentheses.

For example:

```
FEB 15 15:41:22 EVENT#4 FIRED: (Service Toys exceeded 30,000
connections).
```

For information on configuring an RMON event, see the "Configuring an RMON Event" section. For information on configuring an RMON alarm, see the "Configuring an RMON Alarm" section.

Viewing a Traplog File

A traplog file is an ASCII file in the log directory containing generic and enterprise SNMP traps. No configuration is necessary for the traplog file. When an RMON alarm event occurs, a notification of its occurrence is automatically saved in the trap log file on the CSS. Even when traps are disabled, the CSS still produces a log message for any event that would normally generate a trap.

The traps sent to the traplog file are the same traps sent to an SNMP network management station. Refer to Chapter 12, Configuring Simple Network Management Protocol (SNMP) for information on configuring SNMP.

To display all SNMP traps that have occurred on the CSS, enter:

```
# show log traplog
```

Viewing a CSS Disk Log File

Before the CSS can send an event to a log location, you must:

- Configure the location by using the **logging disk**, **host**, **line**, or **sendmail** command.
- Enable logging for the network management subsystem. To do so, enter:

```
(config)# logging subsystem netman level info-6
```

Refer to Chapter 8, Using the CSS Logging Features, for details on configuring logging for the CSS.

To view the events in a log file on the CSS disk, use the **show log** *log_filename* command. To view a log file named log1, enter:

```
# show log log1
```

RMON Configuration in a Startup-Config File

The following example shows an RMON configuration in a startup-config file.

```
!************************ RMON EVENT*************************
rmon-event 1
 active
 description "Service connections exceeded 100"
 owner "Boston Tech Lab"
 community moonbase_alpha
 type log-and-trap

rmon-event 2
 active
 description "Service connections are below 90"
 owner "Boston Tech Lab"
 community moonbase_alpha
 type log-and-trap

!************************* RMON ALARM *************************
rmon-alarm 1
 active
 owner "Boston Tech Lab"
 sample-variable apSvcConnections.
 sample-type absolute
 startup-type rising-and-falling
 rising-threshold 100
 rising-event 1
 falling-threshold 90
 falling-event 1
 sample-interval 30

!************************ RMON HISTORY ************************
rmon-history 5
 active
 owner Boston Tech Lab
 data-source ifIndex.3
 interval 60
 requested-buckets 25
```

Upgrading Your CSS Software

New software versions are periodically released for the CSS. This appendix provides information on how to upgrade your CSS with a new software release. This appendix contains the following major sections:

- Before You Begin
- Upgrading Your CSS Software
- Updating MIBs

Note When syntax changes are made to existing CLI commands, the CSS updates your startup-config file automatically with most command syntax changes. For example, the CSS automatically updates the **web-mgmt state enabled** command in the startup-config file to the new **no restrict web-mgmt** command. If the CSS does not update a command syntax change in a startup-config file automatically, a startup error is displayed. Refer to the *Release Note for the Cisco Series Content Services Switch* for information on which command syntax changes display startup-config file errors.

Before You Begin

Before you can upgrade your CSS, you must copy the new CSS software to your FTP server and configure an FTP server record so the CSS recognizes the server. Use the **show installed-software version-limit** command to display the maximum number of installed versions allowed on your hard disk or Flash disk.

Copying the New CSS Software

ArrowPoint Distribution Images (ADIs) of the CSS software versions are located on the Cisco Systems Web site (www.cisco.com). Use your customer login and password to access this page. From this location, access the page listing the versions of GZIP-compressed software, then click an image to download it. Once the image is downloaded, place it on an FTP server that the CSS can access.

Note You do not need to uncompress the GZIP-compressed software. When you copy the software, or if the upgrade script copies the software to the CSS, the CSS automatically uncompresses it.

Configuring an FTP Server Record on the CSS

Before you can copy the ADI from the FTP server to the CSS, you must create an FTP record file on the CSS identifying the ADI. The record contains the IP address, username, and password for the server. To configure an FTP server record:

1. Log in to the CSS.
2. Access global configuration mode:

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

3. Use the **ftp-record** command to configure the default FTP server. The syntax is:

   ftp-record *ftp_record ip_or_host username*
   [“*password*”|**encrypted-password** *encrypted_pwd*] {*base_directory*}

   The options and variables for this command are:

   - *ftp_record* - Name for this FTP record file. Enter an unquoted text string with no spaces and a maximum of 32 characters.
   - *ip_or_host* - IP address or host name of the FTP server you want to access. Enter an IP address in dotted-decimal notation (for example, 192.168.11.1) or a mnemonic host name (for example, myhost.mydomain.com).
   - *username* - Valid login username on the FTP server. Enter a case-sensitive unquoted text string with no spaces and a maximum of 32 characters.
   - *password* - Password for the valid login username on the FTP server. Enter a case-sensitive, quoted text string with no spaces and a maximum of 16 characters. The CSS allows all special characters in a password except for the percent sign (%).
   - **encrypted-password** *encrypted_pwd* - Encrypted password for the valid login username on the FTP server. Enter a case-sensitive, unquoted text string with no spaces and a maximum of 16 characters.
   - *base_directory* - (Optional) Base directory when using this record.

   For example:

   ```
   (config)# ftp-record DEFAULT_FTP 192.168.2.01 eng1
   encrypted-password serve
   ```

You can now upgrade your CSS.

Upgrading Your CSS Software

This section includes the following topics:

- Using the Upgrade Script
- Manually Upgrading the CSS Software

Using the Upgrade Script

The upgrade script allows you to upgrade your CSS without having to enter any CLI commands. There are two ways to run the script:

- Automatically Running the Upgrade Script
- Interactively Using the Upgrade Script

If the upgrade script fails while upgrading the CSS to the same version of software that is currently running, the CSS software directory will be incomplete. To reinstall the software, you must upgrade the CSS manually (that is, use FTP to transfer the .adi to the CSS and perform a manual unpack).

Automatically Running the Upgrade Script

You can run the upgrade script to perform the software upgrade without having to enter any information. The script automatically:

- Checks to see how many installed software versions are installed on the CSS. You can install a maximum of two software versions. If the CSS contains the maximum number of installed software versions, the script deletes an older version.

Note The script does not prompt you to delete a version of software that you configured as the primary or secondary boot file. On a Flash disk-based system, you may need to quit and then deselect the primary or secondary boot file before continuing with the upgrade.

- Archives the running configuration to the startup configuration.
- Copies the new ADI to the CSS boot-image directory.
- Unpacks the new ADI.
- Sets the primary boot file to the new ADI.
- Reboots the CSS.

To upgrade your CSS software using the upgrade script:

1. Log in to the CSS.
2. Archive your custom scripts and user-profile files from the CSS scripts directory to the archive directory. The upgrade overwrites files in the script directory but does not overwrite files in the archive directory. After the upgrade, you will restore these files to the scripts directory.

   To archive each file to the archive directory, use the **archive script** command. The syntax for this command is:

   archive script *script_filename* {*archive_filename*}

   The variables are as follows:

   - *script_filename* - The filename of the script to archive. To see a list of scripts, enter:

     ```
     # archive script ?
     ```

   - *archive_filename* - (Optional) The name you want to assign to the archive file. Enter an unquoted text string with a maximum of 32 characters.

   For example, to archive the admin-profile file from the scripts directory to the archive directory, enter:

   ```
   # archive script admin-profile
   ```

   To copy any changes to your current user profile to the script directory and then archive the profile to the archive directory, use the alias **save_profile** command. For example, enter:

   ```
   # save_profile
   ```

3. Start the upgrade script and put the name of the ADI and its extension in quotes.

   - If you are using a GZIP-compressed ADI from the FTP server, include the **.gz** file extension. For example:

     ```
     # upgrade “ap0720002.adi.gz”
     ```

   - If you are using an uncompressed version of the ADI from the FTP server, include the **.adi** file extension. For example:

     ```
     # upgrade “ap0720002.adi”
     ```

   If you did not configure a default FTP record before starting the upgrade script, the script prompts you to configure one. You can either:

   - Allow the CSS to automatically configure a record to the server containing the ADI
   - At the prompts, manually configure the FTP record by entering the FTP server information where you copied the upgrade ADI

   When you configure a default FTP record, you see the following information during the upgrade:

   ```
   Current Version:ap0720002 (Build 2)

   *** You must remove an installed version to upgrade. ***

   Attempting to delete ap0720001

   archive running-config startup-config

   Attempting ftp of ap0720002.adi:
   #     copy ftp DEFAULT_FTP ${new_version_adi} boot-image
   Copying (-) 57,241,012
   Completed successfully.
   #(config-boot)#
   unpack ${new_version_adi}
   Unpacking(/) 99%
   (config-boot)#
   setting primary boot-file ap0720002

   rebooting
   ```

   The CSS automatically performs a Flash upgrade, if necessary, and then boots the new image.

4. After you upgrade the software in a CSS 11506 that contains a passive SCM, use the **passive sync** command in boot-config mode (or the **passive sync** macro command) immediately after upgrading your CSS software to synchronize the boot configurations on the redundant SCMs. Refer to Chapter 4, Specifying the CSS Boot Configuration for details on configuring a boot configuration record for a passive SCM.

5. Use the **restore** command to restore the startup-config file, custom scripts, and user-profile files previously archived in the CSS archive directory. To see a list of files in the archive directory, enter:

   ```
   # restore ?
   ```

   - To restore the startup-config file, use the **restore** *filename* **startup-config** command. For example, to restore the startup-config file in the archive directory as the startup-config file on the CSS, enter:

     ```
     # restore startup-config startup-config
     ```

   - To restore each custom script and user profile file to the script directory, use the **restore** *filename* **script** command. For example, to restore the admin-profile filename to the CSS script directory, enter:

     ```
     # restore admin-profile script
     ```

Interactively Using the Upgrade Script

The upgrade script allows you to enter information and make selections by responding to prompts as it runs. Before the script performs the upgrade, it prompts you to:

- Remove ADIs from the CSS if the script detects two installed versions on a hard disk-based system or on a Flash disk-based system
- Enter the version of the new ADI
- Set the primary boot-file to the new ADI
- Reboot the CSS with the ADI you are installing after the upgrade is done
- Archive the running configuration to the startup configuration

To use the interactive version of the script:

1. Log in to the CSS.
2. Archive your custom scripts and user-profile files from the CSS scripts directory to the archive directory. The upgrade overwrites files in the script directory but does not overwrite files in the archive directory. After the upgrade, you will restore these files to the scripts directory.

   To archive each file to the archive directory, use the **archive script** command. The syntax for this command is:

   archive script *script_filename* {*archive_filename*}

   The variables are as follows:

   - *script_filename* - The filename of the script to archive. To see a list of scripts, enter:

     ```
     # archive script ?
     ```

   - *archive_filename* - (Optional) The name you want to assign to the archive file. Enter an unquoted text string with a maximum of 32 characters.

   For example, to archive the admin-profile file from the scripts directory to the archive directory, enter:

   ```
   # archive script admin-profile
   ```

   To copy any changes to your current user profile to the script directory and then archive the profile to the archive directory, use the alias **save_profile** command. For example, enter:

   ```
   # save_profile
   ```

3. Start the upgrade script. For example:

   ```
   # upgrade
   ```

   If you did not configure a default FTP record before starting the upgrade script, the script prompts you to configure one. You can either:

   - Allow the CSS to automatically configure a record to the server containing the ADI
   - At the prompts, manually configure the FTP record by entering the FTP server information where you copied the upgrade ADI

When a default FTP record is configured, the script displays the current version of the ADI.

```
Current Version: ap0720002 (Official)
```

If the script detects the maximum number of ADIs, a message informs you to remove an ADI. Then the script prompts you to remove an older ADI. For example:

```
*** You must remove an installed version to upgrade.***

remove ap0720001[y n q]?
```

Note The script does not prompt you to delete a version of software that you configured as the primary or secondary boot file. On a Flash disk-based system, you may need to quit and then deselect the primary or secondary boot file before continuing with the upgrade.

4. Remove the ADI, if necessary.
   - Enter **y** to remove the displayed ADI version.
   - Enter **n** for the script to display another version to remove.
   - Enter **q** to exit from the script.

```
remove ap0720001 [y n q]? y

Attempting to delete ap0720001
```

5. Enter the filename and extension of the GZIP-compressed ADI version to install at the prompt, and verify the information you entered. If you are using an uncompressed version of the ADI from the FTP server, include the **.adi** file extension (for example, ap0720002.adi). For example:

```
Please Enter Version to Install:ap0720002.adi.gz

Upgrade to Version ap0720002? [y n q] y
```

6. Determine whether to set the ADI as the primary boot file.
   - Enter **y** to set the ADI as the primary boot-file and change the CSS configuration.
   - Enter **n** to keep the same primary boot-file configuration.

```
Set primary boot-file to Version ap0720002? [y n q] y
```

7. Determine whether to have the CSS reboot with the ADI.
   - Enter **y** to reboot the CSS with this ADI after the upgrade is done.
   - Enter **n** to not reboot the CSS with the ADI after the upgrade is done.

   ```
   Reboot with Version ap0720002? [y n q] n
   ```

8. Determine whether to have the CSS archive the contents of the running-config file to the startup-config file.
   - Enter **y** to archive the contents of the running-config file to the startup-config file.
   - Enter **n** to keep the same startup configuration.

   ```
   Archive running-config to startup-config? [y n q] y

   archive running-config startup-config
   ```

   The script copies the ADI from the FTP server, unpacks and installs it, and sets the ADI as the primary boot file.

   ```
   Attempting ftp of ap0720002.adi.gz:

   #      copy ftp DEFAULT_FTP ${new_version_adi} boot-image

   Copying (-) 57,241,012

   Completed successfully.
   #
   (config-boot)# unpack ${new_version_adi}

   unpacking(/) 99%

   (config-boot)#

   setting primary boot-file ap0720002
   ```

9. If you decided to reboot the CSS with the installed ADI in Step 7, the CSS reboots automatically. If you made the ADI the primary boot file and archived the contents of the running-config file to the startup-config file, the CSS automatically performs a Flash upgrade, if necessary, and then boots the new image.

   To manually reboot the system, enter the following commands:

   ```
   (config)# boot
   (config-boot)# reboot
   ```

10. After you upgrade the software in a CSS 11506 that contains a passive SCM, use the **passive sync** command in boot-config mode (or the **passive sync** macro command) immediately after upgrading your CSS software to synchronize the boot configurations on the redundant SCMs. Refer to Chapter 4, Specifying the CSS Boot Configuration for details on configuring a boot configuration record for a passive SCM.

11. Use the **restore** command to restore the startup-config file, custom scripts, and user-profile files previously archived in the CSS archive directory. To see a list of files in the archive directory, enter:

```
# restore ?
```

- To restore the startup-config file, use the **restore** *filename* **startup-config** command. For example, to restore the startup-config file in the archive directory as the startup-config file on the CSS, enter:

```
# restore startup-config startup-config
```

- To restore each custom script and user profile file to the script directory, use the **restore** *filename* **script** command. For example, to restore the admin-profile filename to the CSS script directory, enter:

```
# restore admin-profile script
```

Manually Upgrading the CSS Software

If desired, you can manually enter CLI commands to upgrade the CSS. Make sure you configure a default FTP server, as described in the "Before You Begin" section.

To manually upgrade the software version on your CSS:

1. Log in to the CSS.

2. Remove an older version of the ADI from the CSS if there are two installed versions.

Caution Do not remove the ADI currently running on the CSS. Use the **version** command to see the currently running software version.

To remove an ADI:

a. List the ADIs on the CSS.

```
(config)# show installed-software
ap0720001
ap0720002
```

b. Access boot mode:

```
(config)# boot
(config-boot)#
```

c. Use the **remove** command to remove the ADI.

```
(config-boot)# remove ap0720001
```

3. Archive your running configuration to the startup config file. For example:

```
# config
(config)# archive running-config startup-config
```

You can also use the **save_config** alias to archive your startup-config file. To view all available aliases, use the **show aliases** command.

4. Archive your custom scripts and user-profile files from the CSS scripts directory to the archive directory. The upgrade overwrites files in the script directory but does not overwrite files in the archive directory. After the upgrade, you will restore these files to the scripts directory.

To archive each file to the archive directory, use the **archive script** command. The syntax for this command is:

archive script *script_filename* {*archive_filename*}

The variables are as follows:

- *script_filename* - The filename of the script to archive. To see a list of scripts, enter:

  ```
  # archive script ?
  ```

- *archive_filename* - (Optional) The name you want to assign to the archive file. Enter an unquoted text string with a maximum of 32 characters.

For example, to archive the admin-profile file from the scripts directory to the archive directory, enter:

```
# archive script admin-profile
```

To copy any changes to your current user profile to the script directory and then archive the profile to the archive directory, use the alias **save_profile** command. For example, enter:

```
# save_profile
```

5. Copy the new ADI to the CSS as the boot-image. If you are copying an uncompressed version of the ADI from the FTP server, include the .adi file extension (for example, ap0720002.adi).

```
(config-boot)# <Ctl-z>
# copy ftp DEFAULT_FTP ap0720002.adi.gz boot-image
```

   DEFAULT_FTP is the FTP record file defined in the "Configuring an FTP Server Record on the CSS" section.

   Note When you copy a GZIP-compressed ADI onto the CSS, the CSS automatically uncompresses it.

6. Unpack the ADI.

```
(config)# boot
(config-boot)# unpack ap0720002.adi
```

7. Set the new ADI as the primary boot file and install it. For example:

```
(config-boot)# primary boot-file ap0720002
```

8. Reboot the system.

```
(config)# boot
(config-boot)# reboot
```

   The CSS automatically performs a Flash upgrade, if necessary, and then boots the new image.

9. After you upgrade the software in a CSS 11506 that contains a passive SCM, use the **passive sync** command in boot-config mode (or the **passive sync** macro command) immediately after upgrading your CSS software to synchronize the boot configurations on the redundant SCMs. Refer to Chapter 4, Specifying the CSS Boot Configuration for details on configuring a boot configuration record for a passive SCM.

10. Use the **restore** command to restore the startup-config file, custom scripts, and user-profile files previously archived in the CSS archive directory. To see a list of files in the archive directory, enter:

```
# restore ?
```

- To restore the startup-config file, use the **restore** *filename* **startup-config** command. For example, to restore the startup-config file in the archive directory as the startup-config file on the CSS, enter:

```
# restore startup-config startup-config
```

- To restore each custom script and user profile file to the script directory, use the **restore** *filename* **script** command. For example, to restore the admin-profile filename to the CSS script directory, enter:

```
# restore admin-profile script
```

Updating MIBs

We recommend that you update the CSS Enterprise MIBs after you upgrade the CSS software. CSS Enterprise MIBs are included in the CSS GZIP file. During the software upgrade, the MIBs are loaded into the CSS /mibs directory.

To update the CSS MIBs on your management station after you upgrade the CSS:

1. Use FTP to transfer the MIBs from the CSS MIBs (/v1 or /v2) directory to your management station.
2. Load the MIBs into the management application.

Refer to Chapter 12, Configuring Simple Network Management Protocol (SNMP) for information on CSS Enterprise MIBs.

Using the Offline Diagnostic Monitor Menu

During the boot process, you can access the Offline Diagnostic Monitor (Offline DM) Main menu. The Offline DM Main menu allows you to perform the following functions:

- Set the boot configuration:
  - Configure a primary and secondary location from which the CSS accesses the boot image
  - Configure an IP address for the CSS
  - Configure a subnet mask
  - Configure a default gateway
- Show the boot configuration
- Select Advanced Options to:
  - Delete a software version from the disk
  - Set a password for the Offline DM Main menu
  - Set an administrative username and password
  - Reformat the disk and perform a check disk
  - Configure disks
- Reboot the CSS

This appendix contains the following major sections:

- Accessing the Offline DM Main Menu
- Using the Boot Configuration Menu
- Displaying the Boot Configuration
- Using the Advanced Options

Accessing the Offline DM Main Menu

To access the Offline DM Main menu:

1. Connect a console to the console port on the CSS. Configure the console with the following default values: 9600 baud, no parity, 8 data bits, 1 stop bit, and flow control set to none.
2. Power on the CSS. After the CSS begins to boot (approximately 15 seconds), it displays the following message (for approximately 5 seconds):

   ```
   Would you like to access the Offline Diagnostic Monitor
   menu?(y<cr>)
   ```

   At this point in the boot sequence, you may either:

   - Take no action (or press **n**) and let the CSS continue booting automatically with the default boot configuration.
   - Press **y**, then press **Enter** to halt the boot process and display the Offline DM Main menu.

The Offline DM Main menu appears:

```
MAIN MENU

Enter a menu number:

1* Set Boot Configuration
2. Show Boot Configuration
3* Advanced Options
4. Reboot System
```

An asterisk (*) next to a menu option indicates that the option contains a submenu.

Note If the 5-second time period elapses before you press **y** to halt the boot process, power down the CSS and then power it up again to access the Offline DM menu.

Table B-1 describes each menu item.

Table B-1 Offline Diagnostic Monitor Menu Options

Menu Option	Function
1. Set Boot Configuration	1. Set Primary Boot Configuration 2. Set Secondary Boot Configuration 3. Set IP Address and Subnet Mask r. Return to previous menu
2. Show Boot Configuration	Display boot configurations (including primary and secondary boot configurations, records, and IP information)

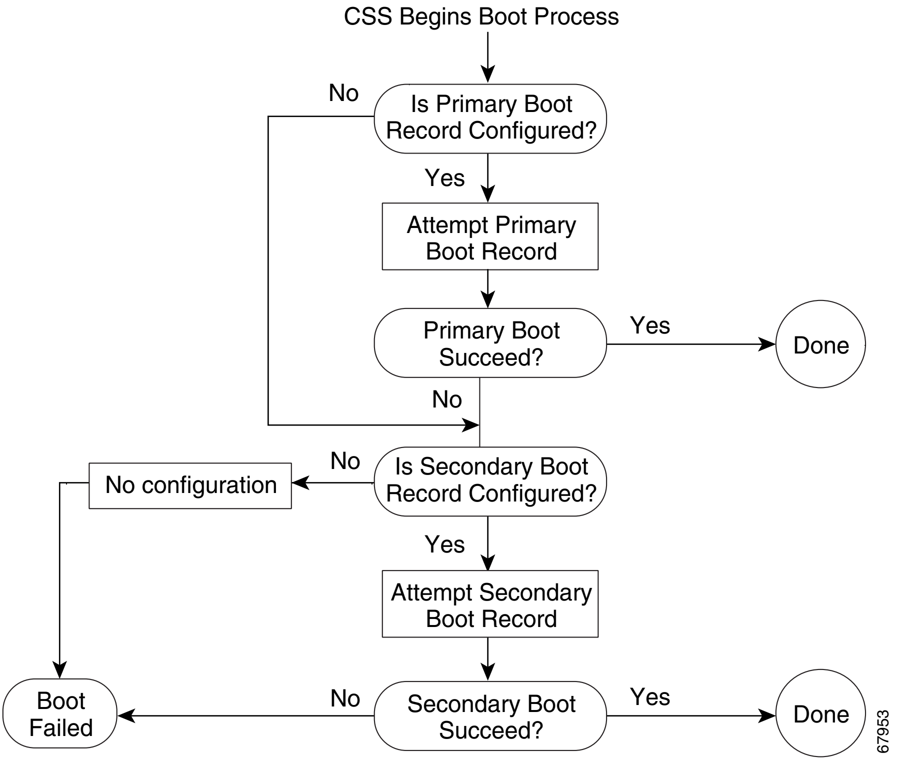
Table B-1 Offline Diagnostic Monitor Menu Options (continued)

Menu Option	Function
3. Advanced Options	1. Delete a software version 2. Security Options 3. Disk Options 4. Set MSD Mapping (two-disk CSS) r. Return to previous menu
4. Reboot System	Reboot the CSS. The CSS displays the following message before rebooting: `Are you sure you want to reboot? (Y/N)` Enter: • **Y** to reboot the CSS • **N** to continue using the Offline DM Main menu

Using the Boot Configuration Menu

The flowchart in Figure B-1 illustrates how the CSS uses the boot configuration information to complete the boot process.

Figure B-1 Boot Configuration Flowchart



The Boot Configuration menu is shown below:

```
BOOT CONFIGURATION MENU

Enter the number of a menu selection:

1. Set Primary Boot Configuration
2. Set Secondary Boot Configuration
3. Set IP Address and Subnet Mask
r. Return to previous menu
```

The Boot Configuration menu enables you to perform the tasks described in Table B-2.

Table B-2 Boot Configuration Options

Menu Option	Function
1. Set Primary Boot Configuration	Specifies the primary location (Network, FTP, Disk, or Clear) from which the CSS accesses the boot image. The default location is Disk.
2. Set Secondary Boot Configuration	Specifies the secondary location (Network, FTP, Disk, or Clear) from which the CSS accesses the boot image. The default location is Clear.
3. Set IP Address and Subnet Mask	Configures an IP address for the Ethernet management port and configure a subnet mask.
r. Return to previous menu	Displays the Offline DM main menu.

Setting Primary Boot Configuration

The information you provide for the primary boot configuration specifies the location from which the CSS accesses the primary boot image upon system reboot or when you download new software. When you select **Set Primary Boot Configuration** from the Boot Configuration menu, the CSS displays the following information. If you have previously entered information, the CSS displays the existing information and default values in square brackets [ ].

```
Configuring PRIMARY Boot Record
Boot via [N]etwork, [F]TP, [D]isk, or [C]lear: [D]
Press <Enter> to continue...
```

- Boot via Network - Allows you to boot the CSS using FTP from CSS software on a network-mounted file system on a remote system
- Boot via FTP - Allows you to download an ADI file containing CSS software that you want to install on the CSS disk (hard or Flash disk)
- Boot via Disk - Allows you to boot the CSS from software currently on the CSS disk (hard or Flash disk)
- Boot via Clear - Instructs the CSS to boot the CSS from the secondary boot record

This section includes the following topics:

- Specifying a Network-Mounted File System as the Primary Boot Record
- Specifying FTP as the Primary Boot Record
- Specifying Disk as the Primary Boot Record
- Specifying Clear as the Primary Boot Record

Specifying a Network-Mounted File System as the Primary Boot Record

Set **Network** as the primary boot record when you want to boot the CSS from a network-mounted file system on a remote system using FTP. Instead of the CSS disk, the network file system contains the CSS software. The CSS boots from this file system and loads the configuration into memory.

Perform a network boot when:

- You want multiple CSSs to use the same boot image while keeping their own configuration information. You provide an alternate path for the location of the configuration information. However, this information must exist on the same network file system with the boot image.

  Note When using an alternate configuration path, make sure the path leads to a directory containing the script, log, and info subdirectories. These subdirectories must contain the files in the corresponding subdirectories in the boot image. First, create these subdirectories on the FTP server, then copy the files from the boot image to the subdirectories.

- The CSS has a disk failure. A network boot allows the CSS to boot independently from its disk and to load the configuration into memory.

Before the CSS can boot from the network:

- Locate the remote system on the network where you want to copy the CSS software.
  - Make sure the CSS can access the system using FTP.
  - Copy the CSS software .zip file from www.cisco.com onto the designated network server.
  - Create a directory and unzip the file into it. This directory will contain all of the boot files and directories.
- On the CSS, create an FTP record to the directory containing the CSS software on the network drive.

- Make sure you properly connect the Ethernet management port on the CSS to the network. Note the locations of the Ethernet management port on the CSS as listed below.
  - CSS 11501 - Front panel 10 Mbps-Ethernet management port
  - CSS 11503 and CSS 11506 - SCM 10 Mbps-Ethernet management port
- Be aware of the following network boot restrictions:
  - A network boot is not supported on UNIX workstations.
  - The War-FTP daemon is not supported for network-booting the system software.

When you select **Network**, the CSS prompts you for the FTP kernel information.

1. Enter the FTP kernel path information. This path is the FTP daemon-addressable location where the boot image has been unpacked. You must also include the FTP server IP address and the username and password to access the boot image. For example:

```
Enter the FTP Kernel path:[] k:/ap0720002/hdd
Enter FTP Server IP address:[] 10.3.6.58
Enter FTP Server authentication username:[] mandy
Enter FTP Server authentication password:[] fred
```

2. Enter an alternate path to the configuration files, including the startup configuration and script files, if the configuration information is not in the same directory as the boot image.

Note The CSS must be able to access the configuration path through the previously configured FTP server IP address, login username, and password.

For example:

```
Enter the FTP Config Path? [] k:/atlanta-config/
Press <Enter> to continue...
```

3. Press **Enter** to display the Boot Configuration menu.
4. Enter **r** to display the Offline DM Main menu.

5. Select option **4** to reboot the CSS.

When the CSS completes the current boot process, it:

- Accesses the network file system containing the boot image
- Boots the CSS using the boot image you specified

Specifying FTP as the Primary Boot Record

Select **FTP** as the primary boot record when you want to upgrade the CSS software on the CSS disk. The CSS accesses the ADI or GZIP file containing the CSS software from an FTP server, copies it to the disk, and unpacks the software. Then the CSS boots from disk (hard or Flash disk).

Make sure that you properly connect the Ethernet management port on the CSS to the network. Note the locations of the Ethernet management port on the CSS as listed below.

- CSS 11501 - Front panel 10 Mbps-Ethernet management port
- CSS 11503 and CSS 11506 - SCM 10 Mbps-Ethernet management port

When you select **FTP**, the CSS prompts you for the boot image filename and FTP information.

Note The CSS FTP server supports only the active (normal) FTP mode of operation. The FTP server does not support the passive FTP mode of operation.

1. Enter a valid FTP pathname, if required. For example:

```
Enter the boot image filename: /ftpimages/ap0720002
Enter FTP Server IP address: 10.3.6.58
Enter FTP Server authentication user name: mandy
Enter FTP Server authentication password: fred
```

The CSS queries if you want to access the boot image directly from the disk at the next reboot (that is, the next time you reboot the CSS after completing the current boot process).

```
Boot from Disk at next reboot? y/n
Press <Enter> to continue...
```

2. Enter one of the following:
    - **y** to copy the boot image from the FTP server to the disk. The CSS accesses the boot image directly from the disk at the next reboot. The CSS also changes the information in the primary boot record to Disk.
    - **n** to FTP the boot image from the FTP server at the next reboot.
3. Press **Enter** to display the Boot Configuration menu.
4. Enter **r** to display the Offline DM Main menu.
5. Select option **4** to reboot the CSS.

When the CSS completes the current boot process, it:

- Accesses the ADI file from the FTP server and unpacks (uncompresses) the file
- Boots the CSS using the boot image you specified

Specifying Disk as the Primary Boot Record

When you select **Disk** as the primary boot record, the CSS displays all boot image versions that reside on the disk. For example:

```
ap0720001
ap0720002
```

1. Enter the boot image filename you wish to use at the prompt.

```
Enter the boot image filename: ap0720002
```

2. Press **Enter** to display the Boot Configuration menu.

```
Press <Enter> to continue...
```

3. Press **r** to display the Offline DM Main menu.
4. Select option **4** to reboot the CSS. Upon reboot, the CSS boots up using the boot image you specified.

Specifying Clear as the Primary Boot Record

To use the secondary boot record information instead of the primary boot record to boot the CSS:

1. Select **Clear** as the primary boot record.
2. Press **Enter** to display the Boot Configuration menu.

3. Press **r** to display the Offline DM Main menu.
4. Select option **4** to reboot the CSS. Upon reboot, the CSS uses the secondary boot record.

Setting Secondary Boot Configuration

The information you provide for the secondary boot configuration specifies the location from which the CSS accesses the boot image if you specified **Clear** as a primary boot record or if the primary boot record fails.

When you select **Set Secondary Boot Configuration** from the Boot Configuration menu, the CSS displays the following information. If you have previously entered information, the CSS displays the existing information and default values in square brackets [ ].

```
Configuring SECONDARY Boot Record
Boot via [N]etwork, [F]TP, [D]isk, or [C]lear: [D]
Press <Enter> to continue...
```

- Boot via Network - Allows you to boot the CSS using FTP from CSS software on a network-mounted file system on a remote system
- Boot via FTP - Allows you to download an ADI file containing CSS software that you want to install on the CSS disk
- Boot via Disk - Allows you to boot the CSS from software currently on the CSS disk
- Boot via Clear - Instructs the CSS to boot the CSS from the primary boot record

This section includes the following topics:

- Specifying a Network-Mounted File System as the Secondary Boot Record
- Specifying FTP as the Secondary Boot Record
- Specifying Disk as the Secondary Boot Record
- Specifying Clear as the Secondary Boot Record

Specifying a Network-Mounted File System as the Secondary Boot Record

Select **Network** as the secondary boot record when you want to boot the system from a network-mounted file system on a remote system using FTP. Instead of the CSS disk, the network file system contains the CSS software. The CSS boots from this file system and loads the configuration into memory. Perform a network boot when:

- You want multiple CSSs to use the same boot image while keeping their own configuration information. You provide an alternate path for the location of the configuration information. However, this information must exist on the same network file system with the boot image.

Note When using an alternate configuration path, make sure the path leads to a directory containing the script, log, and info subdirectories. These subdirectories must contain the files in the corresponding subdirectories in the boot image. First, create these subdirectories on the FTP server, then copy the files from the boot image to the subdirectories.

- The CSS has a disk failure. A network boot allows the CSS to boot independently from its disk and to load the configuration into memory.

Before the CSS can boot from the network:

- Locate the remote system on the network where you want to copy the CSS software.
  - Make sure the CSS can access the system using FTP.
  - Copy the CSS software .zip file from www.cisco.com onto the designated network server.
  - Create a directory and unzip the file into it. This directory will contain all of the boot files and directories.
- On the CSS, create an FTP record to the directory containing the CSS software on the network drive.

- Make sure you properly connect the Ethernet management port on the CSS to the network:
  - CSS 11501 - Front panel 10 Mbps-Ethernet management port
  - CSS 11503 or CSS 11506 - SCM 10 Mbps-Ethernet management port
- Be aware of the following network boot restrictions:
  - A network boot is not supported on UNIX workstations.
  - The War-FTP daemon is not supported for network-booting the system software.

When you select **Network**, the CSS prompts you for the FTP kernel information.

1. Enter the FTP kernel path information. This path is the FTP daemon addressable location where the boot image has been unpacked. You must also include the FTP server IP address and the username and password to access the boot image. For example:

```
Enter the FTP Kernel path:[] k:/ap0720001/hdd
Enter FTP Server IP address:[] 10.3.6.58
Enter FTP Server authentication username:[] mandy
Enter FTP Server authentication password:[] fred
```

2. Enter an alternate path to the configuration files, including the startup-config and script files, if the configuration information is not in the same directory as the boot image.

Note The CSS must be able to access the configuration path through the previously configured FTP server IP address, login username, and password.

For example:

```
Enter the FTP Config Path? [] k:/atlanta-config/
Press <Enter> to continue...
```

3. Press **Enter** to display the Boot Configuration menu.
4. Enter **r** to display the Offline DM Main menu.
5. Select option **4** to reboot the CSS.

When the CSS completes the current boot process, it:

- Accesses the network file system containing the boot image

- Boots the CSS using the boot image you specified

Specifying FTP as the Secondary Boot Record

Select **FTP** as the secondary boot record value when you want to upgrade the CSS software on the CSS disk. The CSS accesses the ADI or GZIP file containing the CSS software from an FTP server, copies it to the disk, and unpacks the software. Then the CSS boots from disk (hard or Flash disk).

Make sure you properly connect the Ethernet management port on the CSS to the network:

- CSS 11501 - Front panel 10 Mbps-Ethernet management port
- CSS 11503 or CSS 11506 - SCM 10 Mbps-Ethernet management port

When you select **FTP**, the CSS prompts you for the boot image filename and FTP information.

1. Enter a valid FTP pathname, if required. For example:

   ```
   Enter the boot image filename: /ftpimages/ap0720002
   Enter FTP Server IP address: 10.3.6.58
   Enter FTP Server authentication user name: mandy
   Enter FTP Server authentication password: fred
   ```

   The CSS queries if you want to access the boot image directly from the disk at the next reboot (that is, the next time you reboot the CSS after completing the current boot process).

   ```
   Boot from Disk at next reboot? y/n
   ```

2. Enter one of the following:
   - **y** to copy the boot image from the FTP server to the disk. The CSS accesses the boot image directly from the disk at next reboot. The CSS also changes the information in the secondary boot record to Disk.
   - **n** to FTP the boot image from the FTP server at next reboot.
3. Press **Enter** to display the Boot Configuration menu.

   ```
   Press <Enter> to continue...
   ```

4. Enter **r** to display the Offline DM Main menu.
5. Select option **4** to reboot the CSS.

   When the CSS uses the secondary boot record to reboot, it:

- Accesses the ADI file from the FTP server and unpacks (uncompresses) the file
- Boots the CSS using the boot image you specified

Specifying Disk as the Secondary Boot Record

When you select **Disk** as the secondary boot record, the CSS displays all boot image versions that reside on the disk and prompts you to enter a boot image.

1. Enter a boot image filename.

   ```
   Boot via [N]etwork, [F]TP, [D]isk, or [C]lear: [D]

   ap0720001
   ap0720002

   Enter the boot image filename: ap0720001
   ```

2. Press **Enter** to display the Boot Configuration menu.

   ```
   Press <Enter> to continue...
   ```

3. Enter **r** to display the Offline DM Main menu.
4. Select option **4** to reboot the CSS. Upon reboot, the CSS boots up using the boot image you specified.

Specifying Clear as the Secondary Boot Record

If you do not wish to specify a secondary boot record:

1. Select **Clear** as the secondary boot record.
2. Press **Enter** to display the Boot Configuration menu.
3. Enter **r** to display the Offline DM Main menu.
4. Select option **4** to reboot the CSS. Upon reboot, the CSS uses the primary boot record.

Setting IP Address, Subnet Mask, and Default Gateway

When you select **Set IP Address and Subnet Mask** from the Boot Configuration menu, the CSS prompts you to:

1. Enter an IP address for the Ethernet management port.

   An IP address of 0.0.0.0 for the Ethernet management port is a legal setting and disables the management port upon reboot. If you enter 0.0.0.0, and you attempt to use the **subnet mask** command, the following message appears: `The mask cannot be set because the IP address is 0.0.0.0`. To enable the Ethernet management port, use the Boot Configuration menu to enter an IP address for the management port.

   The Ethernet management port IP address must be a different subnet than any other CSS VLAN circuit IP subnet. If you do not make the Ethernet management port IP address unique, you cannot access the port.

   ```
   Enter IP Address or 0.0.0.0 to disable [<current value>]:
   10.3.6.58
   ```

2. Enter a subnet mask.

   ```
   Enter Subnet Mask: 255.0.0.0
   ```

3. Enter a default gateway address for the Ethernet management port.

   ```
   Enter Default Gateway: 172.16.11.1
   ```

4. Press **Enter** to display the Boot Configuration menu.

   ```
   Press <Enter> to continue...
   ```

5. Enter **r** to display the Offline DM Main menu.
6. Select option **4** to reboot the CSS.

Displaying the Boot Configuration

When you select **Show Boot Configuration** from the Offline DM Main menu, the CSS displays the following boot information. The Miscellaneous information appears only if you set password-protection on the Offline DM Main menu.

```
***************** Miscellaneous ********************
Offline Diagnostic Monitor menu is password-protected
***************** IP/MAC Information ***************
IP Address:10.3.6.58
Subnet Mask:255.0.0.0
Gateway Address:172.16.11.1
MAC Address00-10-58-00-12-ca
***************** PRIMARY **************************
Boot Type:DISK
Boot File:ap0720002
***************** SECONDARY ************************
Boot Type: DISK
Boot File: ap0720001
```

1. Press **Enter** to display the Offline DM Main menu.

   ```
   Press <Enter> to continue...
   ```

2. Press **r** to display the Offline DM Main menu.

Using the Advanced Options

The CSS hard disk or Flash disk enables you to store two versions of software (including the version you are currently running). If you are storing the maximum number of software versions on the CSS and wish to download a new version to the disk, you must delete a version before the CSS allows the download to begin.

When you select **Advanced Options** from the Offline DM Main menu, the CSS displays the Advanced Options menu:

```
A D V A N C E D   O P T I O N S

Enter the number of a menu selection:

1. Delete a Software Version
2* Security Options
3* Disk Options
4. Set MSD Mapping
r. Return to previous menu
```

Deleting a Software Version

To delete a software version from the disk:

1. Select option **1** to display the software versions currently stored on the disk. The CSS prompts you to enter the software version to delete. For example:

```
ap0720001
ap0720002

Enter the software version to delete: ap0720001
```

Warning **Ensure you do not delete the software version that you are currently running.**

2. Press **Enter** to display the Advanced Options menu.

```
Press <Enter> to continue...
```

3. Enter **r** to display the Offline DM main menu.
4. Select option **4** to reboot the CSS.

Using the Security Options

The Security Options menu enables you to:

- Set password protection on the Offline DM menu
- Change the administrative username and password

The Security Options menu is as follows:

```
S E C U R I T Y  O P T I O N S

Enter the number of a menu selection:

1. Set Password Protection for Offline Diagnostic Monitor
2. Set Administrative Username and Password
r. Return to previous menu
```

Setting Password Protection

The CSS allows you to password-protect the Offline DM Main menu against unauthorized access. The default is disabled; no password is required to access the Offline DM Main menu.

Caution Use care when password protecting the Offline DM Main menu and ensure you take adequate steps to record the password. If you lose the new password, it cannot be recovered and you cannot access the Offline DM Main menu. At that point, the only solution would be to contact the Cisco Technical Assistance Center (TAC) at 1-800-553-2447 or 1-408-526-7209. You can also e-mail TAC at tac@cisco.com.

To access the Offline DM Main menu password protection option:

1. Select option **1** from the Security Options menu.

   ```
   Password protect Offline Diagnostic Monitor menu (yes,no):
   The administrative username and password are required to access
   the Offline Diagnostic Monitor menu.
   ```

   - If you enter **yes**, the CSS prompts you to enter a username and password when you access the Offline DM Main menu.
   - If you enter **no**, the CSS does not prompt you for a username and password when you access the Offline DM Main menu.

2. Press **Enter** to display the Security Options menu.

   ```
   Press <Enter> to continue...
   ```

3. Enter **r** to return to the Advanced Options menu.
4. Enter **r** to return to the Offline DM Main menu.
5. Enter **4** to reboot the CSS, or select another option to continue using the Offline DM Main menu.

Changing the Administrative Username and Password

During the initial log in to the CSS, you enter the default user name **admin** and the default password **system** as lowercase text. For security reasons, you should change the administrative username and password through the Security Options menu. Security on your CSS can be compromised because the administrative username and password are configured in the same way for every CSS.

Note You may also use the **username-offdm** CLI command to change the default administrative username and password (see Chapter 2, Configuring CSS Basics).

The administrative username and password are stored in nonvolatile random access memory (NVRAM). Each time you reboot the CSS, it reads the username and password from NVRAM and reinserts them into the user database. SuperUser status is assigned to the administrative username by default.

You can change the administrative username and password, but because the information is stored in NVRAM, you cannot permanently delete them. If you delete the administrative username using the **no username** command, the CSS deletes the username from the running-config file, but restores the username from NVRAM when you reboot the CSS.

To change the administrative username and password through the Offline DM Main menu:

1. Select option **2** from the Security Options menu.
2. Enter a username. The CSS prompts you for this username when you log in. The CSS also prompts you for this username and password if you set password protection on the Offline DM Main menu.

   ```
   Enter <administrator> username (minimum 4 characters):
   ```

3. Enter a password. Note that the CSS does not display passwords.

   ```
   Enter <administrator> password:
   ```

4. Reenter the password for confirmation.

   ```
   Confirm <administrator> password:
   ```

   The CSS displays the Security Options menu.
5. Enter **r** to return to the Advanced Options menu.
6. Enter **r** to return to the Offline DM Main menu.
7. Enter **4** to reboot the CSS, or select another option to continue using the Offline DM Main menu.

Using the Disk Options

The Disk Options menu includes the following menu items:

- **Format Disk** - Reformats the disk. This option permanently erases all data on the disk. If you wish to retain the startup configuration, be sure to move it off the CSS before reformatting the disk. Also make sure you have a copy of the CSS software ADI file to reinstall on the CSS.
- **Check Disk** - Runs a quick check disk or a complete check of the disk.
- **Check Disk Disable** - Disables running a check of the disk at boot time or enable it again. By default, check disk is enabled.

Note We do not recommend running a Flash disk with the **Check Disk Disable** option selected.

The Disk Options menu is as follows:

```
D I S K   O P T I O N S

Enter the number of a menu selection:

1. Format Disk
2. Check Disk
3. Check Disk Disable
r. Return to previous menu
```

Reformatting the Disk

If the CSS detects unrecoverable errors when performing a disk check, you must reformat the disk. Reformatting the disk erases all data from the disk permanently.

To reformat the disk:

1. Select option **1** from the Disk Options menu. If the CSS contains two disks, you see the following prompt:

   ```
   Format volume in which PCMCIA slot? (0,1):
   ```

2. Enter **0** (for slot 0) or **1** (for slot 1).
3. Press **Enter** to continue.

   The CSS prompts you to format the disk.

   ```
   Formatting the disk results in all disk data being permanently
   erased.
   Are you sure you want to continue? (yes,no):
   ```

4. Enter one of the following:
   - **yes** to reformat the disk.
   - **no** to stop the reformat function. If the disk has unrecoverable errors and you do not reformat it, be aware that the file system may be corrupted and functionality compromised.

The CSS prompts you to perform a quick format or a complete format.

```
Quick format? (yes,no):
```

5. Enter one of the following:
   - **yes** to reformat the disk using the quick format (does not perform cluster verification). Use the quick format only when you are certain of the disk integrity.
   - **no** to reformat the disk including cluster verification.

   After the CSS reformats the disk, it displays:

   ```
   Operation completed successfully.
   ```

6. Enter **r** to return to the Advanced Options menu.
7. Enter **r** to return to the Offline DM Main menu.

   Because the disk is empty, you must configure a primary boot record to instruct the CSS where to locate the new ADI file containing the CSS software.

8. Select option **1** to set the primary boot configuration (see the "Setting Primary Boot Configuration" section).

   If you do not set the primary boot configuration before booting the CSS, the boot process halts at the prompt:

   ```
   Would you like to access the Offline Diagnostic Monitor
   menu?(y<cr>)
   ```

   You must enter the Offline DM Main menu to set the primary boot configuration.

Performing a Check Disk

During the check disk process, the CSS detects and recovers from the following error conditions:

- File Allocation Tables (FATs) are out of synchronization
- Sector write truncation revitalization (may occur from a power loss at the time the CSS is writing to the disk)
- Bad cluster identification (of all in-use cluster) and mapping in the FAT when reformatting the disk
- Crosslinked FAT entries
- Disk entry validation, name, size, cluster assignment, cluster chaining
- Recovery of lost clusters

The amount of time the CSS requires to perform a disk check is proportional to the number of installed files and directories on the disk. The greater the number of installed files and directories, the longer the CSS takes to complete the disk check.

Note The CSS cannot recover from sector failures located within the first 754 sectors on the disk (for example, boot, primary/secondary FAT, or root directory entries). If a sector failure occurs, use the Format Disk option of the Disk Options menu (see the "Reformatting the Disk" section). If the CSS is unable to properly format the CSS disk after using the Format Disk option, contact TAC.

To perform a disk check:

1. Select option **2** from the Disk Options menu. The CSS prompts you about correcting errors if discovered.

   ```
   Correct errors if discovered (yes,no):
   ```

2. Choose whether you want the CSS to correct errors it detects. Enter one of the following:

   - **yes** to enable the CSS to correct recoverable errors it detects. When the CSS completes the disk check, it displays a summary of what was fixed.
   - **no** to prevent the CSS from correcting recoverable errors it detects. The CSS displays a summary of what would have been fixed if you had run the disk check.

   The CSS prompts you to perform a quick check of the disk.

   ```
   Quick check disk? (yes,no):
   ```

3. Choose whether you want the CSS to perform a quick disk check or a complete disk check. Enter either:

   - **yes** to instruct the CSS to perform a quick disk check (does not include cluster verification). Use quick disk check only when you are certain of the disk integrity.
   - **no** to instruct the CSS to perform a complete disk check, including cluster verification.

   The CSS performs the disk check. When completed the CSS displays:

   ```
       c:\ - Volume is OK (\)
       Press <Enter> to continue...
   ```

4. Enter **r** to return to the Advanced Options menu.
5. Enter **r** to return to the Offline DM Main menu.
6. Select option **4** to reboot the CSS.

Disabling or Enabling Disk Check

The Disk Options menu provides an option to disable or enable the running of check disk. When you select this option, it toggles to disable check disk if the option is currently enabled, or to enable check disk if the option is currently disabled.

Note We do not recommend running a Flash disk with the **Check Disk Disable** option selected.

For example, if check disk is currently enabled, to disable it:

1. Select option **3** from the Disk Options menu.
2. Enter **r** to return to the Advanced Options menu.
3. Enter **r** to return to the Offline DM Main menu.
4. Select option **2** to display the boot configuration.

   When check disk is disabled, the CSS displays the following:

```
****************** Miscellaneous **********************
Check Disk is disabled
***************** IP/MAC Information ***************
IP Address:    10.3.6.58
Subnet Mask:   255.0.0.0
Gateway Address:172.16.11.1
MAC Address:   00-10-58-00-12-ca
***************** PRIMARY **************************
Boot Type:     DISK
Boot File:     ap0720002
***************** SECONDARY ************************
Boot Type:     DISK
Boot File:     ap0720002
Press <Enter> to continue...
```

If check disk is currently disabled, to reenable it:

1. Select option **3** from the Disk Options menu.
2. Enter **r** to return to the Advanced Options menu.
3. Enter **r** to return to the Offline DM Main menu.
4. Select option **2** to display the boot configuration.

When check disk is enabled, no state information appears in the Miscellaneous field of the boot configuration:

```
***************** IP/MAC Information ***************
IP Address:        10.3.6.58
Subnet Mask:       255.0.0.0
MAC Address:       00-10-58-00-12-ca
***************** PRIMARY **************************
Boot Type:         DISK
Boot File:         ap0720002
***************** SECONDARY ************************
Boot Type:         DISK
Boot File:         ap0720001
Press <Enter> to continue...
```

Configuring Disks in a Two-Disk CSS

The CSS 11501 and the SCM in the CSS 11503 and CSS 11506 contain two PCMCIA slots for a hard disk or Flash disk. These disks contain the CSS system software and are also used for logging and storing offline system files. The two disks are identified by the PCMCIA slots (slot 0 and slot 1) in which they are installed. Disk 0 is the default storage location for the primary and secondary boot records in the CSS. The default storage location for log files and core dumps in the CSS is the specified disk from which the CSS boots (disk 0 or disk 1).

To specify the CSS disk that is to be the storage location for booting (primary and secondary boot disk), for logging output, and for core dumps:

1. Select option **4** from the Advanced Options menu to configure the disks in the active SCM.

   The CSS prompts you to specify the disk for the primary boot record.

   ```
   Set Primary-Boot to which PCMCIA slot? (0,1):
   ```

2. Enter either **0** for the disk in slot 0 (the default setting) or **1** for the disk in slot 1.

   The CSS prompts you to specify the disk for the secondary boot record.

   ```
   Set Secondary-Boot to which PCMCIA slot? (0,1):
   ```

3. Enter either **0** for the disk in slot 0 (the default setting) or **1** for the disk in slot 1.

   The CSS prompts you to specify the disk for core dump files when the CSS experiences a fatal error.

   ```
   Set Core to which PCMCIA slot? (0,1):
   ```

4. Enter either **0** for the disk in slot 0 or **1** for the disk in slot 1.

   Note Core dump information is intended for Cisco Technical Assistance Center (TAC) use only.

   The CSS prompts you to specify the disk for writing information to log files.

   ```
   Set Log to which PCMCIA slot? (0,1):
   ```

5. Enter either **0** for the disk in slot 0 or **1** for the disk in slot 1.
6. Enter **r** to return to the Advanced Options menu.
7. Enter **r** to return to the Offline DM Main menu.
8. Select option **4** to reboot the CSS.

Rebooting the CSS

To reboot the CSS from the Offline DM Main menu:

1. Select option **4** to reboot the system. The following reboot confirmation is displayed:

   ```
   Are you sure you want to reboot? (y/n)
   ```

2. Enter either **y** to reboot or **n** to continue using the Offline DM Main menu.

When the CSS completes the current boot process, it:

- Accesses the network file system containing the boot image
- Boots the CSS using the boot image you last specified

Troubleshooting the Boot Process

There are three phases in the boot process during which the Cisco 11500 series CSS runs power-on self tests on the hardware and checks the boot configuration. During any of these phases, the CSS reports problems through error messages.

- With the CSS 11501, the internal motherboard boots all components in the chassis and verifies that each component is properly functioning.
- With the CSS 11503 and CSS 11506, the SCM boots each module in the chassis and verifies that the module is functioning properly.

This appendix contains the following major sections:

- Diagnostic Tests for Hardware and Error Messages
- Offline DM Verification of the Boot Configuration Record and Disk
- CSS 11501 Boot and Verification
- CSS 11503 and CSS 11506 Boot and Module Verification

If the suggestions in this appendix do not help to resolve your booting problem, contact the Cisco Technical Assistance Center (TAC).

For details about powering on and booting the CSS, including the various bootstates and the Status LEDs, refer to Chapter 1, Booting, Logging In, and Getting Started.

Diagnostic Tests for Hardware and Error Messages

At the beginning of the boot process, the Cisco 11500 series CSS performs diagnostic tests on the hardware. When the CSS powers up, it first displays a series of messages (see the Chapter 1, Booting, Logging In, and Getting Started) and then the hardware goes through a series of power-on self tests.

If an error occurs during a test, the console displays an error message, increments the detected error counter, and continues to the next test until the CSS completes all of the power-on self tests. The error messages appear in the following format:

```
>>>>>>>>FAILURE_START
>
>From: Slot Slot_number, CPU Cpu_number
>Level: Failure_level
>Type: Failure_type
>Major Error ID: Maj_Error_id
>Minor Error ID: Min_Error_id
>Test Ref #: Test_reference
>Test: 'Test_name'
>Details:
>
>Failure_details
>
>>>>>>>>>FAILURE_END
```

Table C-1 lists the fields in the error message and describes their meanings. This information may be useful when in contact with TAC about a specific error message.

Table C-1 Fields in the Diagnostic Monitor Error Message

Field	Description
Slot_number	The slot number reporting the error.
Cpu_number	The CPU number reporting the error. This field is 1 for boards with a single MIPS CPU.
Failure_level	There are three types of failure levels: • Board - The CSS 11501 motherboard or a specific module in the CSS 11503 or CSS 11506. If the CSS completes the boot process, but a component or module has failed, the CSS also generates a boot log message. • Backplane - An EEPROM failure is a catastrophic failure. Contact TAC for technical assistance. • Chassis - A fan failure has occurred. After the boot process has completed, a log message appears with information on which fan has failed. For information on troubleshooting a fan failure, see the *Cisco 11500 Series Content Services Switch Hardware Installation Guide*.

Table C-1 Fields in the Diagnostic Monitor Error Message (continued)

Field	Description
Failure_type	One of four types of failure, Hardware/Fatal, Hardware/Non-Fatal, Software/Fatal, and Software/Non-Fatal. • Fatal errors indicate that a CSS 11501 component or a specific module in the CSS 11503 or CSS 11506 cannot perform its intended function. • Non-Fatal errors indicate that a CSS 11501 component or a specific module in the CSS 11503 or CSS 11506 is capable of performing its intended function despite the errors, but you should repair the problem as soon as possible. In the case of fatal and non-fatal errors with the CSS 11501, contact TAC for technical assistance. In the case of fatal and non-fatal errors with the CSS 11503 or CSS 11506: **1.** Power down the CSS when the CSS completes the boot process. **2.** Reseat the failed module. **3.** Power up the CSS. If reseating the module does not correct the failure, contact TAC for technical assistance.
Maj_Error_id	The single reference number that points to a particular sub-function in the CSS 11501 chassis or a specific module in the CSS 11503 or CSS 11506.
Min_Error_id	The sub-reference number that points to a particular verification step within the sub-function.
Test_reference	The test number associated with a particular test.

Table C-1 Fields in the Diagnostic Monitor Error Message (continued)

Field	Description

Test_name	Provides the name of the test reporting the error. For example:
	`Uart Interrupt Test`
	`PHY Reset Test`

Failure_details	Provides information about the error. For example:
	`PHY Reset Register failed to clear. Addr: 0x12345678`
	`Expected: 0x0 Actual:0xf`

After the CSS performs the diagnostics, it boots the Offline DM as indicated by the following message:

```
Booting OffDm @ 0xbfd70000
```

See the "Offline DM Verification of the Boot Configuration Record and Disk" section for the Offline DM verification of the boot configuration record and disk drive.

If the Booting OffDm message does not appear, a CSS 11501 component failure or an SCM failure may have occurred; such a failure would not allow a software download to start.

If this problem occurs for a CSS 11501, contact Cisco Technical Assistance Center (TAC) for technical assistance.

If this problem occurs for a CSS 11503 or CSS 11506:

1. Power down the CSS.
2. Reseat the SCM.
3. Power up the CSS.

If reseating the module does not correct the failure, contact TAC for technical assistance.

Offline DM Verification of the Boot Configuration Record and Disk

During the Offline DM verification phase, the CSS checks the configuration record and initializes the disk. If the CSS detects any errors in the configuration record, a failed message appears along with information on the configuration parameter in question. The problems may include a misconfigured IP and subnet address, or there is no primary or secondary boot record. The CSS does not continue the boot process until the problem is resolved.

If a failed message occurs:

1. Enter the Offline DM menu and display your current configuration record. Refer to Appendix B, Using the Offline Diagnostic Monitor Menu for detailed information on using Offline DM.
2. Reconfigure the CSS boot record configuration.
3. Reboot the CSS.

Note If a MAC address error occurs, contact TAC for technical assistance.

After the CSS confirms a valid configuration record, it initializes the disk in slot 0. If the disk cannot initialize, the CSS indicates that it has failed. If an initialization problem occurs:

1. Enter the Offline DM menu.
2. Select the option **3** from the Disk Options menu.
3. Perform a check disk on the disk in slot 0. If necessary, reformat the disk.
4. Reboot the CSS. If the failure is not resolved, contact TAC for technical assistance.

CSS 11501 Boot and Verification

After the CSS 11501 completes the Offline DM boot process, the CSS displays the login banner and starts the Online Diagnostic Monitor (OnDM). During OnDM, the CSS 11501 downloads software to each component and verifies that each component is functioning.

If there is a component failure, the CSS 11501 attempts the boot process three times. If the boot is unsuccessful, the CSS generates the following log message and saves the message in the boot.log file:

```
CHMGR: Slot slot/subslot had diagnostic failures - NOT STARTING UP
```

If this problem occurs for a CSS 11501, contact TAC for technical assistance.

CSS 11503 and CSS 11506 Boot and Module Verification

After the CSS 11503 or CSS 11506 completes the Offline DM boot process, the CSS displays the main banner and starts the Online Diagnostic Monitor (OnDM). During OnDM, the SCM downloads software to each of the modules and boots the modules. The SCM verifies that each module is functioning.

If there is a module failure, the SCM attempts to boot the module three times. If the SCM is unsuccessful, the CSS generates the following log message and saves the message in the boot.log file:

```
CHMGR: Slot slot/subslot had diagnostic failures - NOT STARTING UP
```

The SCM disables the slot and no longer recognizes it. If you use the **show chassis** command, the slot does not appear. If a module failure occurs:

1. Power down the CSS.
2. Reseat the module.
3. Power on the CSS.

If reseating the module does not correct the failure, replace the module.

For additional information on troubleshooting the modules during normal CSS operation, see the *Cisco 11500 Series Content Services Switch Hardware Installation Guide*.

INDEX