



## 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](#)



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 SNMP-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

The CSS requires a File Transfer Protocol (FTP) record file to access an FTP server from the CSS. A few examples of uses for an FTP record with the CSS include:

- Copy an ADI, script file, or startup configuration file from an FTP server to the CSS
- Copy a running-configuration file, startup-configuration file, log file, script, or a core dump file from the CSS to an FTP or TFTP server
- Define a keepalive method in which the CSS logs in to an FTP server
- Import or export certificates and private keys from or to a Cisco 11500 series CSS
- Associate an FTP access mechanism with a service for demand-based replication activities
- Write a portion or all of the Proximity Database to a file in the log directory on the CSS disk or a file on an FTP server

Use the **ftp-record** command to create the FTP record file. 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|\btftp ip_or_host]filename
running-config]
```

The options and variable for this command are as follows:

- **ftp** *ftp\_record* - Copies the startup-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

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

When you use this command, the CSS checks all services, source groups, content rules, ACLs, the management port, the 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

To format and create the Core and Archive directories on a specified disk, use the **format** command. 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

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). 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, the logging output file, and core dumps.

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

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). Use the **map primary-boot** command to select the disk that contains the primary boot record of the CSS. 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

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). Use the **map secondary-boot** command to select the disk that contains the secondary boot record of the CSS. 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

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. Use the **map core** command to select the disk that stores core dump files when the CSS experiences a fatal error.

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

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. Use the **map log** command to select the disk on which you want to store 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

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

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

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 running-config** - Archives the running-config file
- **archive log** - Archives a log file
- **archive script** - Archives a script or user-profile 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, custom scripts, and user-profile files 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 restore these files from the archive directory.

This section includes the following topics:

- [Archiving the Startup-Config File](#)
- [Clearing the Archive Directory](#)
- [Archiving the Running-Config File](#)
- [Archiving Scripts](#)



---

**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 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.

---

## 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 or user-profile 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.

## 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

The archive directory resides on the CSS disk (hard or Flash disk) to store logs, scripts, and startup-config files. Use the **restore** command to restore files previously archived in the CSS archive directory.

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 or user-profile 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 or user-profile 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 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 11501, CSS 11503, or CSS 11506. 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:                sg0730002 (7.30.0.02)
Network Path:           e:/adi_directory/
Config Path:            e:/adi_directory/
Flash (Locked):         7.20.0.03
Flash (Operational):   7.30.0.02
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: <ul style="list-style-type: none"> <li>• primary</li> <li>• backup</li> <li>• powered-off</li> <li>• powered-on</li> <li>• bad</li> <li>• unknown</li> <li>• empty slot</li> </ul>
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: <ul style="list-style-type: none"> <li>• Online</li> <li>• Offline</li> </ul>

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: <ul style="list-style-type: none"> <li>• primary</li> <li>• backup</li> <li>• powered-off</li> <li>• powered-on</li> <li>• bad</li> <li>• unknown</li> <li>• empty slot</li> </ul>
Port Number	The Ethernet port number.

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

<b>Field</b>	<b>Description</b>
Port Name	The port name.
Operational Status	The status of the port. The possible states are as follows: <ul style="list-style-type: none"><li>• online</li><li>• offline-ok</li><li>• offline-bad</li><li>• bad</li><li>• going-online</li><li>• going-offline</li><li>• inserted</li><li>• post</li><li>• post-ok</li><li>• post-fail</li><li>• post-bad-comm</li><li>• any</li><li>• unknown-state</li></ul>

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

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

<b>Field</b>	<b>Description</b>
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)*

<b>Field</b>	<b>Description</b>
Status	The operational status of the module. The possible states are as follows: <ul style="list-style-type: none"> <li>• primary</li> <li>• backup</li> <li>• powered-off</li> <li>• powered-on</li> <li>• bad</li> <li>• unknown</li> <li>• empty slot</li> </ul>
Port Number	The Ethernet port number.
Port Name	The port name.
Operational Status	The status of the port. The possible states are as follows: <ul style="list-style-type: none"> <li>• online</li> <li>• offline-ok</li> <li>• offline-bad</li> <li>• bad</li> <li>• going-online</li> <li>• going-offline</li> <li>• inserted</li> <li>• post</li> <li>• post-ok</li> <li>• post-fail</li> <li>• post-bad-comm</li> <li>• any</li> <li>• unknown-state</li> </ul>

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.



### Note

Issuing the **show system-resources** command can cause CSS CPU usage to increase. If the CSS 11503 or 11506 has multiple modules installed, using this command increases CPU usage accordingly. The increased CPU usage is a direct result of the computational overhead that occurs when the CSS polls the modules and calculates CPU usage.

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
<b>Buffer Statistics</b>	
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, 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 the 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 is 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	<p>The type of user. Types are as follows:</p> <ul style="list-style-type: none"> <li>• Administrator (administrative username, created using the <b>username-offdm</b> command)</li> <li>• Technician (technician username, created using the <b>username-technician</b> command)</li> </ul> <p>If the field is blank, the user is neither an administrator nor a technician.</p> <p><b>Note</b> The <b>username-offdm</b> command is for use by system administrative personnel only. The <b>username-technician</b> command is for use by technical personnel only.</p>

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

Field	Description
Directory Access	<p>The directory access privileges for the listed usernames (as specified through the <b>dir-access</b> option of the <b>username</b> 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:</p> <ul style="list-style-type: none"> <li>• R - Read-only access to the CSS directory</li> <li>• W - Write-only access to the CSS directory</li> <li>• B - Both read- and write-access privileges to the CSS directory (default for all users)</li> <li>• N - No access privileges to the CSS directory</li> </ul> <p>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.</p>

## 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** command 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.