

# format

To format bootflash or a Flash PC card (a Flash device must be formatted before it can be used), use the **format** command.

```
format [spare spare-num] [m/]device1: [[device2:][monlib-filename]]
```

| Syntax Description            |  |  |
|-------------------------------|--|--|
| <b>spare</b> <i>spare_num</i> | (Optional) Indicates the number of spare sectors to reserve when other sectors fail.               |  |
| <i>m/</i>                     | (Optional) Module number of the supervisor engine containing the Flash device.                     |  |
| <i>device1</i> :              | Flash device to be formatted.  |  |
| <i>device2</i> :              | (Optional) Flash device that contains the <i>monlib</i> file to be used to format <i>device1</i> . |  |
| <i>monlib-filename</i>        | (Optional) Name of the <i>monlib</i> file.   |  |

**Defaults** The default number of spare sectors is 0.

**Command Types** Switch command.

**Command Modes** Privileged.

**Usage Guidelines**

A colon (:) is required after the specified device.

You can reserve up to 16 spare sectors for use when other sectors fail. If you do not reserve a spare sector and later some sectors fail, you will have to reformat the entire Flash memory, which will erase all existing data.

The *monlib* file is the ROM monitor library used by the ROM monitor to access files in the Flash file system. It is also compiled into the system image. In the command syntax, *device1*: is the device to format and *device2*: contains the *monlib* file to use.

When you omit the [[*device2*:][*monlib-filename*]] argument, the system formats *device1*: using the *monlib* that is bundled with the system software.

When you omit *device2*: from the [[*device2*:][*monlib-filename*]] argument, the system formats *device1*: using the named *monlib* file from the device specified by the **cd** command.

When you omit *monlib-filename* from the [[*device2*:][*monlib-filename*]] argument, the system formats *device1*: using the *monlib* file from *device2*:. When you specify the whole [[*device2*:][*monlib-filename*]] argument, the system formats *device1*: using the specified *monlib* file from the specified device.

You can also specify *device1:monlib-filename* as the device and filename to be used, as follows:

**format device1:** [*device1*: [*monlib-filename*]]

If *monlib-filename* is omitted, the system formats *device1*: using the built-in monlib file on the device.

**Note**

---

When the system cannot find a monlib file, the system terminates the formatting process.

---

**Note**

---

If the Flash device has a volume ID, you must provide the volume ID to format the device. The volume ID is displayed using the [show flash m/device: filesystem](#) command.

---

---

**Examples**

This example shows how to format a Flash PC card:

```
Console> (enable) format slot0:
All sectors will be erased, proceed (y/n) [n]?y
Enter volume id (up to 31 characters):
Formatting sector 1
Format device slot0 completed.
Console> (enable)
```

# frame

To display an individual stack frame, use the **frame** command.

```
frame [-d | -p] [num]
```

| Syntax Description |   |
|--------------------|---|
| <b>-d</b>          | (Optional) Specifies a monitor context.                                     |
| <b>-p</b>          | (Optional) Specifies a booted image process level context.                  |
| <i>num</i>         | (Optional) Number of the frame to display, where <b>0</b> = youngest frame. |

**Defaults** The default is a booted image kernel context, which is the youngest frame.

**Command Types** ROM monitor command.

**Command Types** Normal.

**Usage Guidelines** The minus sign (-) is required with the **-d** and **-p** options.

**Examples** This example shows how to use the **frame** command to specify a booted image process level context, frame 1:

```
rommon 6 > frame -p 1
Stack Frame 1, SP = 0x80007ed8, Size = 32 bytes
[0x80007ed8 : sp + 0x000] = 0x6031de50
[0x80007edc : sp + 0x004] = 0x6031c000
[0x80007ee0 : sp + 0x008] = 0x00000000
[0x80007ee4 : sp + 0x00c] = 0x80007ec4
[0x80007ee8 : sp + 0x010] = 0x00000002
[0x80007eec : sp + 0x014] = 0x00000000
[0x80007ef0 : sp + 0x018] = 0x60008770
[0x80007ef4 : sp + 0x01c] = 0x600087f0
```

# fsck

To check a Flash file system for damage and to repair any problems, use the **fsck** command.

**fsck** [*m*]/*device*: [**automatic**]

| Syntax Description |   |
|--------------------|---|
| <i>m</i>           | (Optional) Number of the module that contains the Flash device.                               |
| <i>device</i> :    | Name of the Flash device; valid device names are <b>disk0:</b> and <b>disk1:</b> .            |
| <b>automatic</b>   | (Optional) Specifies automatic mode. See the “Usage Guidelines” section for more information. |

**Defaults** This command has no default settings.

**Command Types** Switch command.

**Command Modes** Privileged.

**Usage Guidelines** In automatic mode, problems are fixed automatically and you are not prompted to confirm any changes that will be made to the file system.

**Examples** This example shows how to check a file system for damage and to make repairs. First, enter the **dir** command to list files on a device and to display the file that is corrupted:

```
Console> (enable) dir disk0:
  3  -rw-      556   Mar 06 2049 16:26:16 t1
  4  -rw-      556   Mar 06 2049 16:26:16 t2
  5  -rw-      556   Mar 06 2049 16:26:16 t3
  6  -rw-    258048   Mar 06 2049 16:26:16 t4
```

CORRUPTED

```
Console> (enable)
```

```
128090112 bytes available (16384 bytes used)
```

Then, enter the **fsck** command to repair the corrupted file:

```
Console> (enable) fsck disk0:
```

```
Checking the partition table and boot sector...
Checking FAT, Files and Directories...
File size of disk0:/t4 is not correct, correcting it
Reclaiming unused space...
Updating FAT...
Console> (enable)
```

Enter the **dir** command again to see that the corrupted file is corrected:

```
Console> (enable) dir disk0:
  3  -rw-      556   Mar 06 2049 16:26:16 t1
  4  -rw-      556   Mar 06 2049 16:26:16 t2
  5  -rw-      556   Mar 06 2049 16:26:16 t3
  6  -rw-     4096   Mar 06 2049 16:26:16 t4
CORRECT
Console> (enable)
```

**Related Commands**    [dir—switch](#)

## history—ROM monitor

To display the command history (the last 16 commands executed in the ROM monitor environment), use the **history** command. This command is aliased to “h” by the ROM monitor for convenience.

### history

**Syntax Description** This command has no arguments or keywords.

**Defaults** This command has no default settings.

**Command Types** ROM monitor command.

**Command Modes** Normal.

**Examples** This example shows how to use the **history** command:

```
rommon 13 > history
```

```
1  help
2  break -s 0x20090
3  break -s 10090
4  break -s 0xa0001000
5  cont
6  help
7  dev
8  dir
9  dir bootflash:
10 dis
11 dis 0xa0001000
12 dis 0xbe000000
13 history
```

```
=====
```

# history—switch

To show the contents of the command history buffer, use the **history** command.

## history

---

**Syntax Description** This command has no arguments or keywords.

---

**Defaults** This command has no default settings.

---

**Command Types** Switch command.

---

**Command Modes** Normal.

---

**Usage Guidelines** The history buffer size is fixed at 20 commands. See the “[Command-Line Interfaces](#)” chapter for detailed information about the command history feature.

---

**Examples** In this example, the **history** command lists the contents of the command history buffer:

```
Console> history
  1 help
  2 history
Console> !2
history
  1 help
  2 history
  3 history
Console>
```

# I2trace

To display the Layer 2 path taken by the packets that start at a specified source address and end at a specified destination address, use the **I2trace** command.

```
I2trace src_mac_addr dest_mac_addr [vlan] [detail]
```

```
I2trace src_ip_addr dest_ip_addr [detail]
```

| Syntax Description   |  |  |
|----------------------|--|--|
| <i>src_mac_addr</i>  | Source MAC address.                        |  |
| <i>dest_mac_addr</i> | Destination MAC address.                   |  |
| <i>vlan</i>          | (Optional) Number of the VLAN.             |  |
| <i>src_ip_addr</i>   | Source IP address or alias.                |  |
| <i>dest_ip_addr</i>  | Destination IP address or alias.           |  |
| <b>detail</b>        | (Optional) Specifies detailed information. |  |

**Defaults** This command has no default settings.

**Command Types** Switch command.

**Command Types** Privileged.

**Usage Guidelines** All the intermediate devices should be Catalyst 5000 or Catalyst 6500 series switches running supervisor engine software release 6.1 or later. Catalyst 4000 family switches must be running supervisor engine software release 6.2 or later.

The **I2trace** command displays the Layer 2 path when the specified source and destination addresses belong to the same VLAN. If you specify source and destination addresses that belong to different VLANs, **I2trace** aborts with an error message.

You must enable CDP on all the Catalyst 4000, Catalyst 5000, or Catalyst 6500 series switches in the network.

When the switch detects a device (in the Layer 2 path) that does not belong to the Catalyst 4000, Catalyst 5000, or Catalyst 6500 series switch, the switch continues to send Layer 2 trace queries and lets them time out.

This command is rejected if you enter a multicast source or destination MAC address.

If a source or the destination address belongs to multiple VLANs, you must specify the VLAN to be used for determining the Layer 2 path.

The Layer 2 trace feature is not supported when multiple devices are attached to one port through hubs (for example, multiple CDP neighbors detected on a port). When more than one CDP neighbor is detected on the port, l2trace is aborted.

If you specify the IP address of the source and destination systems instead of the MAC addresses, the switch looks at the ARP table to determine the IP address to MAC address mapping of the source and destination systems. If an ARP entry exists for the specified IP address, the corresponding MAC address is used. If no matching ARP entry exists, the system does an ARP query and tries to resolve the IP address. If this is the case, a restriction is imposed that requires the source and destination systems to be in the same subnet as the switch in order for the ARP query to be resolved.

## Examples

This example shows how to display the Layer 2 packet path for a specified source and destination MAC address:

```
Console> (enable) l2trace 00-01-22-33-44-55 10-22-33-44-55-66 detail
l2trace vlan number is 10.

00-01-22-33-44-55 found in C5500 named wiring-1 on port 4/1 10Mb half duplex
C5500: wiring-1: 192.168.242.10: 4/1 10Mb half duplex -> 5/2 100MB full duplex
C5000: backup-wiring-1: 192.168.242.20: 1/1 100Mb full duplex -> 3/1-4 FEC attached
C5000: backup-core-1: 192.168.242.30: 4/1-4 FEC attached -> 1/1-2 GEC attached
C6000: core-1: 192.168.242.40: 1/1-2 GEC attached -> 2/1 10MB half duplex.
10-22-33-44-55-66 found in C6000 named core-1 on port 2/1 10MB half duplex.
Console> (enable)
```

This example shows how to display the Layer 2 packet path for a specified source and destination IP alias:

```
Console> (enable) l2trace user-1-pc user-2-pc detail
Mapping IP address to MAC Address
user-1-pc -> 00-01-22-33-44-55
user-2-pc -> 10-22-33-44-55-66
l2trace vlan number is 10

00-01-22-33-44-55 found in C5500 named wiring-1 on port 4/1 10Mb half duplex
C5500: wiring-1: 192.168.242.10: 4/1 10Mb half duplex -> 5/2 100MB full duplex
C5000: backup-wiring-1: 192.168.242.20: 1/1 100Mb full duplex -> 3/1-4 FEC attached
C5000: backup-core-1: 192.168.242.30: 4/1-4 FEC attached -> 1/1-2 GEC attached
C6000: core-1: 192.168.242.40: 1/1-2 GEC attached -> 2/1 10MB half duplex.
10-22-33-44-55-66 found in C6000 named core-1 on port 2/1 10MB half duplex.
Console> (enable)
```

This example shows how to display a summary of Layer 2 packet path information for a specified source and destination IP address:

```
Console> (enable) l2trace 9.7.0.7 9.7.0.6
Starting L2 Trace
sc0 :9.7.0.7 : 3/7
4/16 :9.7.0.2 : 4/10
Console> (enable)
```

This example shows how to display a summary of Layer 2 packet path information for a specified source and destination MAC address:

```
Console> (enable) l2trace 00-01-22-33-44-55 10-22-33-44-55-66  
Starting L2 Trace  
sc0 :9.7.0.7 : 3/7  
4/16 :9.7.0.2 : 4/10  
Console> (enable)
```

# meminfo

To display information about the main memory, packet memory, and NVRAM, use the **meminfo** command. With the **-l** option, the supported DRAM configurations are displayed.

## **meminfo [-l]**

|                           |  |
|---------------------------|--|
| <b>Syntax Description</b> | <b>-l</b> (Optional) Specifies the long listing, which displays the DRAM configurations. |
|---------------------------|--|

|                 |                                       |
|-----------------|---------------------------------------|
| <b>Defaults</b> | This command has no default settings. |
|-----------------|---------------------------------------|

|                      |                      |
|----------------------|----------------------|
| <b>Command Types</b> | ROM monitor command. |
|----------------------|----------------------|

|                      |         |
|----------------------|---------|
| <b>Command Modes</b> | Normal. |
|----------------------|---------|

|                         |   |
|-------------------------|---|
| <b>Usage Guidelines</b> | The minus sign (-) is required with the <b>-l</b> option. |
|-------------------------|---|

|                 |   |
|-----------------|---|
| <b>Examples</b> | This example shows how to use the <b>meminfo</b> command: |
|-----------------|---|

```
rommon 9 > meminfo
```

```
Main memory size: 16 MB in 32 bit mode.  
Available main memory starts at 0xa000e000, size 16328KB  
IO (packet) memory size: 25 percent of main memory.  
NVRAM size: 32KB
```

# ping

To send ICMP echo-request packets to another node on the network, use the **ping** command. You can also use the **ping** command without arguments to configure ping.

**ping -s** *host*

**ping -s** *host* [*packet\_size*] [*packet\_count*]

**ping**

| Syntax Description  |  |  |
|---------------------|--|--|
| <b>-s</b>           | Causes <b>ping</b> to send one datagram per second, printing one line of output for every response received. |  |
| <i>host</i>         | IP address or IP alias of the host.  |  |
| <i>packet_size</i>  | (Optional) Number of bytes in a packet, from 56 to 1472 bytes.   |  |
| <i>packet_count</i> | (Optional) Number of packets to send; valid values are from 0 to 2,147,483,647.                              |  |

## Defaults

The defaults for **ping -s** are as follows:

- *packet\_size* is 56 bytes
- *packet\_count* is 2,147,483,647

The defaults for **ping** with no arguments are as follows:

- *packet\_size* is 56 bytes
- *packet\_count* is 5
- Wait time is 2 seconds
- Target IP address is none (this is a mandatory field)
- Source address is the host IP address

## Command Types

Switch command.

## Command Modes

Normal or privileged.

## Usage Guidelines

General **ping** command guidelines are as follows:

- Press **Ctrl-C** to stop pinging.
- Continuous ping means that, unless you press **Ctrl-C** to stop pinging, packets are generated continually and dispatched to the host.
- The actual packet size is 8 bytes larger than the size you specify because the switch adds header information.
- Normal response—The normal response occurs in 1 to 10 seconds, depending on network traffic.

The guidelines for the **ping -s** command are as follows:

- The maximum waiting time before timing out is 2 seconds.
- A new ping packet is generated after 1 second of sending the previous packet, regardless of whether or not an echo-reply is received.
- If you do not enter a packet count, continuous ping results.
- Network or host unreachable—The switch found no corresponding entry in the route table.
- Destination does not respond—If the host does not respond, a “no answer from host” appears in 2 seconds.
- Destination unreachable—The gateway for this destination indicates that the destination is unreachable.

The guidelines for the **ping** command without arguments are as follows:

- The **ping host** command is accepted in normal mode only. The parameters take the default values automatically.
- The target IP address is a mandatory field to be entered.
- The maximum waiting time is configurable.
- A new ping packet is generated only when an echo-reply is received.
- Entering a packet count of 0 results in continuous ping.
- Returns output only when a response is received or you press **Return**.
- Available in privileged mode only.
- When configuring ping, you must either press **Return** or enter a response. Valid responses and appropriate values are as follows:
  - Target IP address: IP address or host name of the destination node you plan to ping.
  - Number of Packets: Number of ping packets to be sent to the destination address; valid values are from 0 to 2,147,483,647 (0 specifies continuous ping).
  - Datagram size: Size of the ping packet; valid values are from 56 to 1472 bytes.
  - Timeout in seconds: Timeout interval; valid values are from 0 to 3600 seconds.
  - Source IP Address [(default)]: IP address or IP alias of the source.

## Examples

This example shows how to ping a host with IP alias elvis a single time:

```
Console> ping elvis
!!!!

-----172.20.52.19 PING Statistics-----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms) min/avg/max = 1/1/1
Console>
```

This example shows how to ping a host with IP alias elvis once per second until you press **Ctrl-C** to stop pinging:

```
Console> ping -s elvis
ping elvis: 56 data bytes
64 bytes from elvis: icmp_seq=0. time=11 ms
64 bytes from elvis: icmp_seq=1. time=8 ms
64 bytes from elvis: icmp_seq=2. time=8 ms
64 bytes from elvis: icmp_seq=3. time=7 ms
```

```
64 bytes from elvis: icmp_seq=4. time=11 ms
64 bytes from elvis: icmp_seq=5. time=7 ms
64 bytes from elvis: icmp_seq=6. time=7 ms
^C

----elvis PING Statistics----
7 packets transmitted, 7 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 7/8/11
Console>
```

This example shows how to configure ping:

```
Console> (enable) ping

Target IP Address []: 172.20.52.19
Number of Packets [5]: 6
Datagram Size [56]: 75
Timeout in seconds [2]: 1
Source IP Address [172.20.52.18]:
!!!!!!

----172.20.52.19 PING Statistics----
6 packets transmitted, 6 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 1/1/1
Console> (enable)
```

---

**Related Commands**

[set interface](#)  
[set ip route](#)  
[show interface](#)  
[show ip route](#)