

# Cisco AS5200 Universal Access Server ROM Monitor

---

This appendix describes the Cisco AS5200 ROM monitor, the first software to run when the access server is powered-up or reset. The ROM monitor can help you isolate or rule out hardware problems encountered when installing your access server.

This appendix describes:

- Entering the ROM Monitor Program.
- Aliasing ROM Monitoring Commands.
- ROM Monitor Commands.
- Virtual Configuration Register.

## Entering the ROM Monitor Program

The ROM monitor diagnostics help initialize the processor hardware and boot the main operating system software. If you set the software configuration register (bits 3, 2, 1, and 0) to zero, you can start the access server in the stand-alone ROM monitor. An example of the ROM monitor prompt follows:

```
rommon 1 >
```

To enable the Break key, and to default to booting at the ROM monitor while running the system software, reset the configuration register to 0x0 by entering configuration mode, and enter the configuration command following:

**config-reg 0x0**

The new configuration register value, 0x0, takes effect after the access server is rebooted with the **reload** command. If you set the configuration to 0x0, you will have to manually boot the system from the console each time you reload the access server.



**Timesaver** Break (system interrupt) is always enabled for 60 seconds after rebooting the system, regardless of whether break is configured to be off by setting the configuration register. During the 60-second window, you can break to the ROM monitor prompt.

## ROM Monitor Command Conventions

Following are ROM monitor command conventions:

- Brackets [ ] denote an optional field. If a minus option is followed by a colon (for example: [-s:]), you must provide an argument for the option.

- A word in *italics* means that you must fill in the appropriate information.
- All address and size arguments to the memory-related commands are assumed to be hexadecimal (no “0x” prefix or “h” suffix needed).
- The options [-bwl] for the memory-related commands provide for byte, word, and longword operations. The default is word.
- You can invoke the memory-related commands by typing the command with no arguments. This causes the utility to prompt you for parameters. This option is available for the commands marked as prompting.
- All the built-in commands can be aborted (user interrupt signal) by pressing the **Break** key at the console.
- You can place more than one command (except the repeat command) on a line by using the “;” delimiter.

## Aliasing ROM Monitoring Commands

The ROM monitor supports command aliasing modeled on the aliasing function built into the Korn shell. The alias command is used to set and view aliased names. This allows the user to alias command names to a letter or word. Aliasing is often used to shorten command names or automatically invoke command options.

Aliases are stored in NVRAM and remain intact across periods of no power. These are some of the set aliases.

```
b=boot  
h=history  
i=reset  
r=repeat  
k=stack  
?=help
```

## ROM Monitor Commands

Enter **?** or **help** at the *rommon 12 >* prompt to display a list of available commands and options, as follows:

```
rommon 12 > help
alias          set up and display alias
boot           boot up an external process
confreg        configuration register utility
cont           continue executing a downloaded image
context        display the context of a loaded image
dev            list the device table
dir            list files in file system
dnld           serial download a program module
frame          print out a selected stack frame
help           monitor builtin command help
history        monitor command history
meminfo        main memory information
repeat         repeat a monitor command
reset          system reset
set            display the monitor variables
stack          produce a stack trace
sync           write monitor environment to NVRAM
sysret         print out info from last system return
unalias        unset an alias
unset          unset a monitor variable
xmodem         x/y modem download
```

---

**Note** You can display additional details for a command by entering the command name with a **-?** option, which prints the command usage message.

---

The commands are listed and described in alphabetical order. Note that the ROM monitor commands are case-sensitive.

- **alias** [name=value]—Aliases a name to a value. If the value contains white space or other special (shell) characters, it must be quoted. If the value has a space as the last character the next command line word is also checked for an alias (normally only the first word on the command line is checked). Without an argument, this command prints a list of all aliased names with their values.

For example:

```
rommon 1 > alias
r=repeat
h=history
?=help
b=boot
ls=dir
```

- **boot** or **b**—Boots an image. The **boot** command with no arguments boots the first image in boot Flash memory. You can include an argument, *filename*, to specify a file to be booted over the network using the Trivial File Transfer Protocol (TFTP). The local device (see the description of **b device** following) can be specified by entering the device specifier (*devid*). If the specified device name is not recognized by the ROM monitor, the system will attempt to boot the image (*imagename*) from a network TFTP server. Do not insert a space between *devid* and *imagename*. Options to the boot command are **-x**, load image but do not execute, and **-v**, verbose. The form of the **boot** command follows:

```
boot [-xv] [devid] [imagename]
```

**b**—Boots the default system software from ROM.

**b filename [host]**—Boots using a network TFTP server. When a host is specified, either by name or IP address, the boot command will boot from that source.

**b flash:**—Boots the first file in Flash memory.

**b device:**—Boots the first file found in the Flash memory device. The Flash memory device specified can be either *flash:*, to boot the Cisco IOS software, or *bootflash:*, to boot the boot image in Flash memory.

**b device:name**—An extension of the above command, allows you to specify a particular filename in the Flash memory bank.

- **confreg [hexnum]**—Executing the **confreg** command with the argument *hexnum* changes the virtual configuration register to match the hex number specified. Without the argument, **confreg** dumps the contents of the virtual configuration register in English and allows the user to alter the contents. You are prompted to change or keep the information held in each bit of the virtual configuration register. In either case, the new virtual configuration register value is written into NVRAM and does not take effect until you reset or power cycle the access server.

The configuration register resides in NVRAM. The configuration register is identical in operation to other Cisco access servers. Enter **confreg** for the menu-driven system, or enter the new value of the register in hexadecimal.

---

**Note** The value is always interpreted as hexadecimal. The **confreg** utility will print a before and after view of the configuration register when used in menu-driven mode.

---

For example:

```
rommon 7 > confreg
```

```
Configuration Summary
```

```
enabled are:
```

```
console baud: 9600
```

```
boot: the ROM Monitor
```

```
do you wish to change the configuration? y/n [n]: y
```

```
enable "diagnostic mode"? y/n [n]: y
```

```
enable "use net in IP bcast address"? y/n [n]:
```

```
enable "load rom after netboot fails"? y/n [n]:
```

```
enable "use all zero broadcast"? y/n [n]:
```

```
enable "break/abort has effect"? y/n [n]:
```

```
enable "ignore system config info"? y/n [n]:
```

```
change console baud rate? y/n [n]: y
```

```
enter rate: 0 = 9600, 1 = 4800, 2 = 1200, 3 = 2400 [0]: 0
```

```
change the boot characteristics? y/n [n]: y
```

```
enter to boot:
```

```
0 = ROM Monitor
```

```
1 = the boot helper image
```

```
2-15 = boot system
```

```
[0]: 0
```

```
Configuration Summary
```

```
enabled are:
```

```
diagnostic mode
```

```
console baud: 9600
```

```
boot: the ROM Monitor
```

```
do you wish to change the configuration? y/n [n]:
```

```
You must reset or power cycle for new config to take effect.
```

- **cont** [-b]—Continues a loaded image that has stopped. The -b option sets the requested break points before continuing.

For example:

```
reboot >
```

```
monitor: command "launch" aborted due to user interrupt
```

```
diagmon 7 > cont
```

```
reboot>
```

- **context**—Displays the CPU context at the time of the fault. The context from the kernel mode and process mode of a booted image is displayed, if available.

For example:

```
rommon 6 > context
CPU Context:
d0 - 0x00000028      a0 - 0x0ff00420
d1 - 0x00000007      a1 - 0x0ff00000
d2 - 0x00000007      a2 - 0x02004088
d3 - 0x00000000      a3 - 0x020039e6
d4 - 0x00000000      a4 - 0x02002a70
d5 - 0x02003e8a      a5 - 0x02003f17
d6 - 0x00000000      a6 - 0x02003938
d7 - 0x00000001      a7 - 0x0200392c
pc - 0x02004adc      vbr - 0x02000000
```

- **cookie**—Displays the contents of the cookie PROM in hexadecimal format.

For example:

```
rommon 1 > cookie
cookie:
01 01 00 00 0c 07 af 80 07 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

- **dev**—Lists boot device identifications on the access server.

For example:

```
rommon 10 > dev
Devices in device table:
      id  name
eprom: eprom
flash: PCMCIA slot 1
```

- **dir *devid***—Lists the files on the named device.

For example:

```
rommon 11 > dir flash:
      File size      Checksum      File name
      65 bytes (0x41)      0xb49d      cleve/oddfiles65
2229799 bytes (0x220627)  0x469e      C5200-k.z
```

- **dlnd [-xv:] [*args*]**—Downloads in binary format through the console and executes. The -x option downloads but does not execute. The -v option allows you to specify the verbose level. The optional arguments are passed to the downloaded program via the argc/argv mechanism (only when -x is not used). The exit value is the return value from the downloaded routine or the status of the download operation (success or failure) if the -x option is used.
- **frame [*number*]**—Displays an entire individual stack frame. Enter a number to indicate which frame to display. You can also specify a number to indicate which stack frame to display. Note that the default is 0 (zero), which is the youngest frame.

For example:

```
rommon 6 > frame 2
Frame 02: FP = 0x02003960    RA = 0x020050ee
at 0x02003968 (fp + 0x08) = 0x02004f8d
at 0x0200396c (fp + 0x0c) = 0x0200f390
at 0x02003970 (fp + 0x10) = 0x02006afc
at 0x02003974 (fp + 0x14) = 0xc0a82983
at 0x02003978 (fp + 0x18) = 0x02003a7e
at 0x0200397c (fp + 0x1c) = 0x02002630
at 0x02003980 (fp + 0x20) = 0x00000000
at 0x02003984 (fp + 0x24) = 0x02000000
at 0x02003988 (fp + 0x28) = 0x0200c4a4
at 0x0200398c (fp + 0x2c) = 0x0200f448
```

- **help**—The **help** command prints a summary of the ROM monitor commands to the console screen. This is the same output as entering **?**.

For example:

```
rommon 11 > help
alias          set up and display alias
boot           boot up an external process
confreg        configuration register utility
cont           continue executing a downloaded image
context        display the context of a loaded image
dev            list the device table
dir            list files in file system
dnld           serial download a program module
frame          print out a selected stack frame
help           monitor builtin command help
history        monitor command history
meminfo        main memory information
repeat         repeat a monitor command
reset          system reset
set            display the monitor variables
stack          produce a stack trace
sync           write monitor environment to NVRAM
sysret         print out info from last system return
unalias        unset an alias
unset          unset a monitor variable
xmodem         x/y modem download
```

- **history** or **h**—Displays the command history, that is, the last 16 commands executed in the monitor environment.
- **meminfo**—Displays the size (in bytes) the starting address, the available range of the main memory, the starting point and size of packet memory, and the size of nonvolatile memory (NVRAM).

For example:

```
rommon 9 > meminfo

Main memory size: 8 MB. Packet memory size: 4 MB
Available main memory starts at 0xa000e001, size 0x7f1fff
Packet memory starts at 0xa8000000
NVRAM size: 0x20000
```

- **repeat** [*number or string*] [*count*] or **r**—Repeats the specified command. Without an argument, repeats the last command. The optional command number (from the history list) or match string specifies which command to repeat. In the case of the match string, the most recent command to begin with the specified string will be re-executed. If the string includes spaces, you must define it using quotes. The count option allows you to repeat the command more than once.

- **reset** or **i**—Resets and initializes the system, similar to power-on.
- **set**—Displays all the monitor variables and their values.
- **stack** [*num*]—Produces a stack trace of the *num* frames. The default is 5. The command dumps from the kernel stack and the process stack (if one is available) of a booted image.

For example:

```
rommon 5 > stack 8
Stack trace:
PC = 0x02004adc
Frame 00: FP = 0x02003938    RA = 0x02005f2a
Frame 01: FP = 0x02003948    RA = 0x02005df0
Frame 02: FP = 0x02003960    RA = 0x020050ee
Frame 03: FP = 0x02003994    RA = 0x02004034
Frame 04: FP = 0x02003b00    RA = 0x00012ca6
```

- **sync**—Writes the working in-core copy of the environment variables and aliases to NVRAM so that they are read on the next reset.
- **sysret**—Displays the return information from the last booted system image. This includes the reason for terminating the image, a stack dump of up to eight frames, and if an exception is involved, the address where the exception occurred.

For example:

```
rommon 8 > sysret
System Return Info:
count: 19, reason: user break
pc:0x60043754, error address: 0x0
Stack Trace:
FP: 0x80007e78, PC: 0x60043754
FP: 0x80007ed8, PC: 0x6001540c
FP: 0x80007ef8, PC: 0x600087f0
FP: 0x80007f18, PC: 0x80008734
```

- **unalias** *name*—Removes *name* and its associated value from the alias list.
- **unset** *VARNAME*—Removes the variable name from the variable list.
- **xmodem** [- *yc*] <*destination file\_name*>—Downloads a system image to the boot Flash memory over the console port. The -y option performs the download. The -c option performs the download using 16-bit CRC error checking. The xmodem transfer protocol supports a 128-byte block size and the transfer begins with a block number starting at 1, which contains file data. This is the default transfer protocol.
- **ymodem** [- *yc*] <*destination file\_name*>—Downloads a system image to the boot Flash memory over the console port. The -y option performs the download. The -c option is ignored for this transfer protocol. The ymodem transfer protocol supports a 1024-byte block size and the transfer begins with a block number starting at 0, which contains the filename, file length, and creation date.



## Virtual Configuration Register

This section describes the Cisco AS5200 series virtual configuration register, the factory default settings, and the procedures for changing those settings.

The Cisco AS5200 series has a 16-bit virtual register, which is written into the nonvolatile random access memory (NVRAM). Use the processor configuration register information contained in this section to do the following:

- Set and display the configuration register value.
- Force the system into the bootstrap program.
- Select a boot source and default boot filename.
- Enable or disable the Break function.
- Control broadcast addresses.
- Set the console terminal baud rate.
- Load operating software from ROM.
- Enable booting from a Trivial File Transfer Protocol (TFTP) server.

Table A-1 lists the meaning of each of the virtual configuration memory bits.

**Table A-1 Virtual Configuration Bit Meanings**

Bit No.	Hex	Meaning
00–03	0x0000–0x000F	Boot field (see Table A-2)
06	0x0040	Causes system software to ignore nonvolatile memory contents
07	0x0080	OEM bit enabled
08	0x0100	Break disabled
10	0x0400	IP broadcast with all zeros
11–12	0x0800	Console line speed
13	0x2000	Boots default ROM software if network boot fails
14	0x4000	IP broadcasts do not have net numbers
15	0x8000	Enables diagnostic messages and ignores NVRAM contents

## Changing Configuration Register Settings

Some common reasons to modify the value of the virtual configuration register follow:

- Recover a lost password.
- Change the console baud rate.
- Enable or disable the Break function.
- Manually boot the operating system using the **b** command at the ROM monitor prompt.
- Force the router to boot automatically its system image in Flash memory, or boot as per any **boot system** commands that are stored in its configuration file in NVRAM.

---

**Note** If the router finds no boot system commands, it uses the configuration register value to form a filename from which to boot a default system image stored on a network server. (See Table A-3.)

---

To change the configuration register while running the IOS software, follow these steps:

**Step 1** Enter the **enable** command and your password to enter the privileged level, as follows:

```
router> enable
Password:
router#
```

**Step 2** At the privileged-level system prompt (router #), enter the command **configure terminal**. You will be prompted as shown in the following example:

```
router# configure term
Enter configuration commands, one per line.
Edit with DELETE, CTRL/W, and CTRL/U; end with CTRL/Z
```

**Step 3** To set the contents of the configuration register, enter the **config-register value** configuration command where value is a hexadecimal number preceded by 0x (see Table A-3), as in the following:

```
config-register 0xvalue
```

(The virtual configuration register is stored in nonvolatile memory.)

**Step 4** Exit the configuration mode by entering Ctrl-Z. The new value settings will be saved to memory; however, the new settings do not take effect until the system software is reloaded by rebooting the router.

**Step 5** To display the configuration register value currently in effect and the value that will be used at the next reload, enter the show version EXEC command, and the value will be displayed on the last line of the screen display as in the following example:

```
Configuration register is 0x142 (will be 0x102 at next reload)
```

**Step 6** Reboot the router. The new value takes effect. Configuration register changes take effect only when the server restarts, for example, when you switch the power off and on or when you issue a **reload** command from the console.

## Configuring the Boot Field

The lowest four bits of the processor configuration register (bits 3, 2, 1, and 0) form the boot field. (See Table A-2.)

**Table A-2 Explanation of Boot Field (Configuration Register Bits 00–03)**

Boot Field	Meaning
00	Stays at the system bootstrap prompt (ROM monitor) on a reload or power cycle
01	Boots the boot helper image as a system image
02-F	Specifies a default netboot filename Enables default booting from system Flash memory Enables boot system commands that override default netboot filename <sup>1</sup>

1. Values of the boot field are 2–15 in the form *cisco<n>-processor\_name*, where  $2 < n < 15$

The boot field specifies a number in binary. If you set the boot field value to 0, you must have console port access to boot the operating system manually. Boot the operating system by entering the **b** command at the bootstrap prompt as follows:

```
> b [tftp] flash filename
```

Definitions of the various command options follow:

**b**—Boots the default system software from ROM

**b flash**—Boots the first file in Flash memory

**b filename [host]**—Boots over the network using TFTP

**b flash [filename]**—Boots the file (*filename*) from Flash memory

For more information about the **b [tftp] flash filename** command, see the appropriate software publications. If you set the boot field value to a value of 2 through F, and there is a valid system boot command stored in the configuration file, the router boots the system software as directed by that value. (See Table A-3.) If you set the boot field to any other bit pattern, the router uses the resulting number to form a default boot filename for netbooting.

If there are no boot commands in the configuration file, the router attempts to boot the first file in system Flash memory. If no file is found in system Flash memory, the router attempts to netboot a default file whose name is derived from the value of the boot field (for example: *cisco2-4500*). If the netboot attempt fails, the boot helper image in boot Flash will boot up.

If boot commands are in the configuration file, the router software processes each boot command in sequence until the process is successful or the end of the list is reached. If the end of the list is reached without a file being successfully booted, the router will retry the netboot commands up to six times unless the boot default ROM software if netboot fails bit (bit 13 of the virtual configuration register) is set. If bit 13 is set, the system boots the boot helper image found in boot Flash memory without any retries.

In the following example, the virtual configuration register is set to boot the router automatically from Flash memory and to ignore Break at the next reboot of the router:

```
router# configure terminal
Enter configuration commands, one per line.
Edit with DELETE, CTRL/W, and CTRL/U; end with CTRL/Z
config-register 0x102
Ctrl-Z
router#
```

The server creates a default boot filename as part of the automatic configuration processes. To form the boot filename, the server starts with cisco and links the octal equivalent of the boot field number, a dash, and the processor-type name. Table A-3 lists the default boot filenames or actions for the processor.

**Note** A **boot system** configuration command in the router configuration in NVRAM overrides the default netboot filename.

**Table A-3 Default Boot Filenames**

Action/File Name	Bit 3	Bit 2	Bit 1	Bit 0
bootstrap mode	0	0	0	0
ROM software	0	0	0	1
cisco2-4000 or cisco2-4500	0	0	1	0
cisco3-4000 or cisco3-4500	0	0	1	1
cisco4-4000 or cisco4-4500	0	1	0	0
cisco5-4000 or cisco5-4500	0	1	0	1
cisco6-4000 or cisco6-4500	0	1	1	0
cisco7-4000 or cisco7-4500	0	1	1	1
cisco10-4000 or cisco10-4500	1	0	0	0
cisco11-4000 or cisco11-4500	1	0	0	1
cisco12-4000 or cisco12-4500	1	0	1	0
cisco13-4000 or cisco13-4500	1	0	1	1
cisco14-4000 or cisco14-4500	1	1	0	0
cisco15-4000 or cisco15-4500	1	1	0	1
cisco16-4000 or cisco16-4500	1	1	1	0

Action/File Name	Bit 3	Bit 2	Bit 1	Bit 0
cisco17-4000 or cisco17-4500	1	1	1	1

Bit 8 controls the console Break key. Setting bit 8 (the factory default) causes the processor to ignore the console Break key. Clearing bit 8 causes the processor to interpret Break as a command to force the system into the bootstrap monitor, halting normal operation. A Break can be sent in the first 60 seconds while the system reboots, regardless of the configuration settings.

Bit 10 controls the host portion of the Internet broadcast address. Setting bit 10 causes the processor to use all zeros; clearing bit 10 (the factory default) causes the processor to use all ones. Bit 10 interacts with bit 14, which controls the network and subnet portions of the broadcast address.

Table A-4 shows the combined effect of bits 10 and 14.

**Table A-4 Configuration Register Settings for Broadcast Address Destination**

Bit 14	Bit 10	Address (<net> <host>)
Off	Off	<ones> <ones>
Off	On	<zeros> <zeros>
On	On	<nets> <zeros>
On	Off	<net> <ones>

Bit 13 determines the server response to a bootload failure. Setting bit 13 causes the server to load operating software from ROM after five unsuccessful attempts to load a boot file from the network. Clearing bit 13 causes the server to continue indefinitely to attempt loading a boot file from the network. By factory default, bit 13 is cleared to 0.

Bits 11 and 12 in the configuration register determine the baud rate of the console terminal.

Table A-5 shows the bit settings for the four available baud rates. (The factory-set default baud rate is 9600.)

**Table A-5 System Console Terminal Baud Rate Settings**

Baud	Bit 12	Bit 11
9600	0	0
4800	0	1
1200	1	0
2400	1	1

## Enabling Booting from Flash Memory

To enable booting from Flash memory, set bits 3, 2, 1, and 0 to a value between 2 through 15. To specify a filename to boot, enter the system software configuration command **boot system flash filename** in the configuration file.

To enter the configuration mode while in the system software image, enter the **configure** command at the enable prompt as in the example following:

```
Gateway# configure
Configuring from terminal, memory, or network [terminal]? term
Enter configuration commands, one per line.
Edit with DELETE, CTRL/W, and CTRL/U; end with CTRL/Z
boot system flash filename
```

To disable break and enable the boot system flash command, enter the **config-register** command with a value as follows:

```
config-reg 0x102
CTRL/Z
```

If you set the configuration register value to 0x102, as in the example, it is not necessary to enter the **boot system flash** command unless there is more than one image in Flash.