

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 Monitor commands.
- ROM Monitor commands.

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 standalone 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.

Command Aliasing

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 >` prompt to display a list of available commands and options:

```
rommon 12 > help

alias          set and display aliases command
boot          boot up an external process
break        set/show/clear the breakpoint
confreg      configuration register utility
cont         continue executing a downloaded image
context      display the context of a loaded image
cookie       display contents of cookie PROM in hex
dev          list the device table
dir          list files in file system
dis          disassemble instruction stream
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
```

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.

- **break**— Need explanation.

- **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 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      clev/oddfiles65
2229799 bytes (0x220627)  0x469e      C5200-k.z
```

- **dis**—???.

For example:

- **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 and display aliases command
boot           boot up an external process
break         set/show/clear the breakpoint
confreg       configuration register utility
cont          continue executing a downloaded image
context       display the context of a loaded image
cookie       display contents of cookie PROM in hex
dev           list the device table
dir           list files in file system
dis          disassemble instruction stream
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
```

- **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.

