



## Using the ROM Monitor

---

This appendix describes the Cisco uBR905 and Cisco uBR925 cable access routers' ROM monitor, which helps you isolate and troubleshoot possible hardware problems when installing the router. The ROM monitor is the first software to run when the cable access router is powered-on or reset—it is permanently part of the cable access router and is always available, regardless of the release of Cisco IOS software that has been downloaded to the router.

This appendix describes:

- [Entering the ROM Monitor, page C-1](#)
- [Command Conventions, page C-2](#)
- [Upgrading the ROM Monitor Software, page C-2](#)
- [Commands, page C-3](#)



### Caution

Users and system administrators do not need to access the ROM monitor during normal operation of the cable access router. The ROM monitor should be used only by trained service technicians or under the direction of a Cisco TAC engineer. Many of the commands available in the ROM monitor put the router in a diagnostic or non-functional state—do not enter any commands in the ROM monitor unless you thoroughly understand their function and how to reverse their effects so that you can restore the router to normal operations.

---

## Entering the ROM Monitor

The ROM monitor initializes the processor hardware and boots the main operating system software. The ROM monitor version introduced in the Cisco IOS Release 12.1(3)XL is for the Cisco uBR905 cable access router and Cisco IOS Release 12.2(2)XA is for the Cisco uBR925 cable access router. It displays the following:

```
rommon 2 > i
System Bootstrap, Version 12.0(19990506:181223)[sjacobso-jac-xi26 2504] DEVELOPMENT
SOFTWARE
Copyright (c) 1994-1999 Cisco Systems, Inc.
UBR905 platform with 16384 Kbytes of main memory
```

The ROM monitor main memory information is shown below:

```
rommon 3 > meminfo
Main memory size: 32 MB
Available main memory starts at 0x14000, size 16304 KB
I/O (packet) memory size: 512 KB
```

NVRAM size: 16 KB

To default to booting at the ROM monitor while running the system software, reset the configuration register to 0x0 by entering the following Cisco IOS global configuration command:

```
Router(config)# config-reg 0x0
```

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



**Note**

After you have entered the ROM monitor, you can return to the normal boot mode by changing the configuration register value to **0x2102**, using the **confreg 0x2102 ROMMON** command. Then reboot the system using the **reset ROMMON** command.

## 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.
- You can place more than one command (except the repeat command) on a line by using the “;” delimiter.

## Upgrading the ROM Monitor Software

The Cisco uBR905 and Cisco uBR925 cable access routers support both a primary and secondary ROM Monitor (ROMMON). The primary ROMMON is permanently installed, while the secondary ROMMON is upgradable.

After power-on or a hard system reset, the primary ROMMON initially takes control. It then checks for the presence of a secondary ROMMON and if present, verifies that the secondary ROMMON has the correct checksum. If the secondary ROMMON passes the validation tests, the primary ROMMON passes control to the secondary ROMMON, which then performs the power-on self-test and hardware initialization, and then loads and executes the Cisco IOS software image. Otherwise, the primary ROMMON remains in control and continues the boot process.

This approach allows the secondary ROMMON to be safely upgraded when new software is available. If the file transfer is interrupted, or if the new ROMMON software becomes corrupted, the primary ROMMON is still available to boot the cable access router and load the Cisco IOS software image.

To upgrade the secondary ROMMON, use the “**copy tftp rommon:**” command in Privileged EXEC mode. See the following example:

```
Router> enable
Router# copy tftp rommon:
Address or name of remote host []? 192.168.100.172
Source filename []? ubr925-rboot-mz
Accessing tftp://192.168.100.172/ubr925-rboot-mz...
Loading ubr925-rboot-mz from 192.168.100.172 (via cable-modem0): !
WARNING...
```

Do not attempt ROMMON upgrades unless you know what you are doing.  
Writing to ROMMON must not be interrupted.  
Do not reset the cable modem during this operation.  
Do what you can to ensure power to the cable modem is not interrupted.

The cable modem will automatically reloaded after ROMMON upgrade is successfully completed.

```
Do you want to continue?[confirm] yes
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 243260/486400 bytes]
```

If the download of the secondary ROMMON is successful, the cable access router is automatically reloaded to transfer control to the new ROMMON.



### Caution

If the download of the secondary ROMMON is interrupted by a power cycle, reset, or network interruption, the secondary ROMMON will become corrupted. You must then reset the cable access router to allow the primary ROMMON to take control and reboot the system. After the system has rebooted with the primary ROMMON, you can repeat the download of the secondary ROMMON.

## 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
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
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
monstat       display primary and secondary ROMMON status
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/ymodem image 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. The ROM monitor's version of command aliasing is based on the aliasing function built into the Korn shell. Aliasing allows you to abbreviate commands or to set up a command so that it is automatically run with certain options.

If the value contains white space or other special 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).

The **alias** command is used to set new aliases and to view the aliases that are currently defined. For example, to display the currently set aliases, enter the alias command by itself:

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

The following command creates the alias "bf" that performs the "boot from Flash memory" command:

```
rommon 1 > alias bf "b flash:"
```

- **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). You can boot from a specific local device by specifying the device name. 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 (first) system software.
  - **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:** (boots the Cisco IOS software) or **bootflash:** (boots the boot image in Flash memory).
  - **b device:name**—Load and run the specified filename from the specified device.
- **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 nonvolatile memory (NVRAM, also known as Flash memory) and does not take effect until you reset or power cycle the router.

The configuration register resides in Flash memory, and 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 "use all zero broadcast"? 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. For example:

```
reboot> launch
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 router. 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
```

- **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 using the *argc/argv* mechanism (valid 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. You can enter a number to indicate which frame to display, and you can also specify a number to indicate which stack frame to display. 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
```

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

- **monstat**—Displays the status of the primary and secondary ROMMON, whether each is valid, and which version is actively running.

- **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 executed again. 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 copy of the environment variables and aliases to Flash memory 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 ending 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: reset
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.

