

The NeXT ROM Monitor

Portions of the ROM Monitor code are copyright of NeXT Inc. Permission pending.

Monitor Commands

- [Inspect or Modify Configuration Parameters](#)
- [Open Address Register](#)
- [Print Memory Configuration](#)
- [Open Data Register](#)
- [Open Processor Register](#)
- [Open System Register](#)
- [Examine Memory Locations](#)
- [Print Error Codes](#)
- [Eject Optical Disk](#)
- [Continue Execution](#)
- [Boot from Device](#)
- [Open Function Code](#)
- [Set Input Radix](#)
- [Set Hardware Password](#)

NMI Commands

- [Resume Execution](#)
- [Break to Debugger](#)
- [Dump Kernel Message Buffer to Screen](#)
- [Exit to Monitor](#)
- [Sync and Reboot System](#)
- [Sync and Halt System](#)
- [Keep This Window Around](#)
- [Print This List](#)
- [Set or Examine Any Kernel Flag](#)

System Test Diagnostics

- [FPU Tests](#)
- [SCC Channel Tests](#)
- [SCSI Channel Tests](#)
- [Ethernet Channel Tests](#)
- [MO Drive ECC Tests](#)
- [Real-Time Clock Tests](#)
- [Two Bit Graphics Tests](#)
- [System Timer Tests](#)
- [Event Counter Tests](#)
- [Sound-Out Tests](#)

The NeXT ROM Monitor

The NeXT ROM monitor performs system test diagnostics and loads the system files when the power

is turned on or the computer is reset. System files can be loaded from the optical disk, a hard disk, or the computer network, depending on the preference set by the Preferences application. Various animated icons are displayed as the system files are being loaded; this process does not require any user input from the keyboard or mouse.

Other capabilities of the ROM monitor are used less frequently. These are accessed through a command interface that uses the keyboard and a text window on the display.

The following sections describes the NeXT ROM monitor, how to access the ROM monitor, how to use the available ROM monitor commands, and how to resume normal program execution after you're through using the ROM monitor.

Monitor Commands

ROM monitor commands are very useful to creating basic software for the NeXT machine. The ROM monitor automatically starts the computer when the power is turned on or when the computer is reset. The ROM monitor is always available via a NMI from the keyboard. Access to it is via the **Command ~** combination.

The ROM monitor commands can be used to examine and modify memory or hardware peripherals on a very basic level. This is important for anyone developing add-on hardware cards for the computer, developing base software or for tracking down other hardware problems. ROM monitor commands can also be used to boot the computer from devices other than the default boot device. The default boot device can also be changed with the **p** command, although this is usually done from the Preferences application.

There are several situations where the ROM monitor will display a text window containing the NeXT> prompt:

- If there's no default boot device set when the computer is turned on.
- If, while the system is already running, you get to the NMI mini-monitor (by pressing the backquote/tilde key on the numeric keypad while holding down both the left and right Command keys) and then give the **mon** or **halt** command to the nmi> prompt. Earlier versions of the ROM code do not support the nmi mode or include it in the basic NeXT prompt. If you used the **mon** command, the system may be continued by using the ROM monitor command **c**. In this way you can interrupt the system in order to use the ROM monitor, and then continue.

You can type a question mark in response to the NeXT> prompt to display the following list of monitor commands:

Monitor Command Effect

p	Inspect/modify configuration parameters
a [<i>n</i>]	Open address register
m	Print memory configuration
d [<i>n</i>]	Open data register
r [<i>regnum</i>]	Open processor register
s [<i>systemreg</i>]	Open system register
e [<i>lwb</i>] [<i>addrlist</i>] [<i>format</i>]	Examine memory location addr
ec	Print recorded system test error codes
ej [<i>drive #</i>]	Eject optical disk cartridge (default = 0)
c	Continue execution at last pc location
b [<i>device</i>] [(<i>ctrl, unit, part</i>)] [<i>file</i>] [<i>flags</i>]	Boot from device
S [<i>fcode</i>]	Open function code (address space)
R [<i>radix</i>]	Set input radix
P	Set hardware password

These commands are described in the following sections.

Most monitor commands print the current value of a register or parameter and prompt for a new value with a question mark. To enter a new value, type the value and press the Return key. To leave the current value unchanged and skip to the next one (if any), just press the Return key. Type a period to exit the command and leave the current value unchanged.

Numeric values are usually in hexadecimal (base 16) notation if they represent a memory address or data item. The **R** (radix) command can be used to change the input base/radix.

Some values have their bitfields displayed symbolically between angle brackets <> in addition to the numeric value (for example, 2700<trace=0,s,ipl=7>). We are looking for the hardware documentation for the meaning of these values

Inspect or Modify Configuration Parameters

p

lets you inspect and modify several parameters, including the default boot command and the system test diagnostics. The default boot command is normally set with the Preferences application but is included here in case you need to change the default but can't boot the system for some reason. All parameters are stored in nonvolatile memory and will survive if the power is turned off. If the hardware password is set, you may not be able to change the values of the parameters.

Example:

NeXT> **p**

boot command: od? **sd** *Make the SCSI drive the default boot device*

DRAM tests: yes? **n** *Disable the DRAM tests*

perform power-on system test: yes?

sound out tests: no?

SCSI tests: no?

loop until keypress: no?

verbose test mode: no?

boot extended diagnostics: no?

serial port A is alternate console: no?

allow any ROM command even if password protected: no?

allow boot from any device even if password protected: no?

allow optical drive #0 eject even if password protected: no?

Possible values for the boot command parameter are:

en Ethernet

sd SCSI
disk

od Optical
disk

Optional flags may be appended to the boot command as described in the section *Boot from Device*. The boot command may be up to 12 characters long.

The DRAM tests, which check main memory and take the longest amount of time, can be enabled separately from the other system test options with the *DRAM tests parameter*.

If the *perform power-on test* parameter is set to **no**, none of the other diagnostic tests are run. If it's set to **yes**, the three additional parameters determine which tests are performed:

The *sound out tests* parameter causes the sound tests to be performed.

The *SCSI tests* parameter enables more comprehensive testing of the SCSI interface (but only if the SCSI bus is terminated by having a SCSI drive installed either internally or externally).

The *loop until keypress* parameter runs the diagnostics repeatedly until interrupted.

The *verbose test mode* parameter causes the name of each test to be printed as it's run.

The *boot extended diagnostics* parameter allows secondary diagnostics to be booted from the default boot device.

If the ROM monitor detects that the display monitor or keyboard is disconnected when the computer is turned on, the ROM monitor will attempt to communicate with a terminal connected to serial port A if the *alternate console* parameter is set to **yes**.

The final three parameters determine how the ROM monitor may be used if the hardware password is set (see the section *Set Hardware Password* for further information):

- The first parameter allows any ROM monitor command to be used, except the **P** command, if the hardware password is set. This is useful if you want to protect the hardware password from being changed without disabling any other ROM commands.
- The second parameter allows any boot device to be specified with the **b** command if the hardware password is set. This can be used to allow booting from the disks and network while disabling other ROM commands.
- The third parameter determines whether the disk in optical drive 0 can be ejected when the hardware password is set. If the computer is installed in a public access area, it may be desirable to prevent the optical disk from being ejected.

Open Address Register

a [n]

accesses the contents of address registers a0 through a7 of the MC68030 processor.

Example:

```
NeXT> a
```

```
a0: 00001234? 5678 Change a0 from 0x1234 to 0x5678
```

```
a1: 00000000?
```

```
a2: 00000001? . Type a period to exit
```

```
NeXT> a0
```

```
a0: 00001234? 5678 Change a0 from 0x1234 to 0x5678
```

```
NeXT>
```

Print Memory Configuration

m

prints information about the memory installed in memory sockets 0 through 15.

Example:

NeXT> **m**

Memory sockets 0-3 have 4MB SIMMs installed

(0x4000000-0x4400000)

Memory sockets 4-7 have 16MB SIMMs installed

(0x5000000-0x6000000)

Memory sockets 8-11 have 4MB page mode SIMMs installed

(0x6000000-0x63fe000)

Memory sockets 12-15 have no SIMMs installed (0x0-0x0)

The highest address of the last SIMM module will be slightly less than you'd expect because the ROM monitor reserves space there to store its internal information.

Open Data Register

d [n]

accesses the contents of data registers d0 through d7 of the MC68030 processor.

Example:

NeXT> **d**

d0: 00001234? **5678** *Change d0 from 0x1234 to 0x5678*

d1: 00000000?

d2: 00000001? . *Type a period to exit*

NeXT> **d0**

d0: 00001234? **5678** *Change d0 from 0x1234 to 0x5678*

NeXT>

Open Processor Register

r [regname]

accesses the contents of the MC68030 processor registers. Possible values for *regname* are:

pc Program counter
 sr Status register
 usp User stack pointer
 isp Interrupt stack pointer
 msp Master stack pointer
 vbr Vector base register
 sfc Source function code
 dfc Destination function code
 cacr Cache control register
 caar Cache address register

Open System Register

s [*systemreg*]

accesses the contents of the system registers. Possible values for *systemreg* are:

intrstat Interrupt status register
 intrmask Interrupt mask register
 scr1 System control register #1
 scr2 System control register #2

Examine Memory Locations

e [*lwb*] [*addrlist*] [*format*]

lets you examine particular locations in memory.

[*lwb*] Specify **l**, **w**, or **b** to select long, word, or byte length. The default is long.

[*addrlist*] This argument specifies the starting address or list of addresses to cyclically examine. If you do not specify a value for *addrlist*, the examine command uses the most recent value of *addrlist*.

[*format*] This argument controls how the value is printed. It can be any of the standard format types supported by the C language **printf()** library routine. The default, **%x**, prints a number in hexadecimal.

Example:

```
NeXT> e 4000000
```

```
4000000: 0? 12345678 Examine the long at memory location 0x4000000; see that its value is 0x0;
deposit the value 0x12345678
```

```
4000004: 0? . Type a period to exit
```

Print Error Codes

ec

prints the last two system error codes recorded in the nonvolatile RAM.

Eject Optical Disk

ej [drive #]

ejects the optical disk contained in the specified optical disk drive. The default drive is drive 0.

Continue Execution

c

continues execution at the last pc location.

Boot from Device

b [device [(ctrl, unit, part)] [file] [flags]]

boots the default kernel, or whatever kernel is specified by *file*.

Possible values for *device* are:

- en Ethernet
- sd SCSI disk
- od Optical disk

The optional device selectors are:

- ctrl* Controller number (default 0)
- unit* Unit number (default 0)
- part* Partition number (default 0)

For SCSI drives the controller number is the logical target number, not the actual target number of the drive. For example if the first drive is set to be target 0 and the second drive target 5, then controller number 1 refers to the second drive.

Possible values for *flags* are:

- a Ask for the name of the root device
- b Don't process the **rc.boot** file
- s Boot in single-user mode instead of multiuser mode
- i Ask for the name of the init program (the default is **/etc/init**)
- p Automatically reboot after a system panic

You can press any key to stop booting from the Ethernet. To stop booting from any device, hold down the right Command key and press the backquote (Á) key on the numeric keypad.

Example:

NeXT> **b** Boot from the default boot device

NeXT> **bod** Boot from the optical disk

NeXT> **ben** Boot from the Ethernet

NeXT> **bod test** Boot **test** from the optical disk

NeXT> **b -s** Boot in single-user mode

NeXT> **ben mach -as** Boot **mach** in single-user mode and prompt for the root device

NeXT> **bsd(1,0,0)** Boot from the second SCSI drive

Open Function Code

S [*fcode*]

lets you inspect or modify the MC68030 function code (address space) used with the **e** (examine) command. The default code is 5 (supervisor data space).

Possible values for *fcode* are:

- 0 Undefined, reserved
- 1 User data space (UD)
- 2 User program space (UP)
- 3 Undefined, reserved
- 4 Undefined, reserved
- 5 Supervisor data space (SD)
- 6 Supervisor program space (SP)
- 7 CPU space

Example:

```
NeXT> S
```

Function code 5 (SD) Current function code is 5

```
NeXT> S 6
```

Function code 6 (SP) Change function code to 6

Set Input Radix

R [*radix*]

lets you set the input radix. The default value for the input radix is 16. Any numbers typed will be interpreted in the base of the input radix (for example, input radix 16 means numbers are interpreted as base 16).

Set Hardware Password

P

lets you set the hardware password, which allows you to limit the ROM monitor commands that can be used without first supplying the password. If the computer is installed in a public access area it may be desirable to limit the use of the ROM, including which devices the computer can boot from and whether the optical disk can be ejected. If the hardware password is set, the three parameters described in the section ^aInspect or Modify Configuration Parameters^o determine how the ROM commands will be limited. If a protected ROM command is entered, it will require the password to be typed before continuing.

NMI Commands

While the NeXT Computer is running, you can produce a nonmaskable interrupt (NMI) by pressing the backquote key on the keypad while holding down both the left and right Command keys; this causes the NMI mini-monitor to display a window that contains the following prompt:

```
nmi>
```

Although the NMI mini-monitor is not part of the ROM monitor, it is similar. The NMI mini-monitor allows low-level inspection and debugging of the system, whereas the ROM monitor allows low-level inspection of the hardware. The **mon** command can be used to transfer control to the ROM monitor from the mini-monitor.

It's also possible to arrive in the NMI mini-monitor window through a system failure. In this case, the prompt will be *panic*> instead of *nmi*>. The available commands are the same (although if you see the *panic*> prompt, the **continue** command will not work).

You can type a question mark in response to the *nmi*> or *panic*> prompt to display the following list of commands:

Commands are: (can be abbreviated)

<u>continue</u>	resume execution
<u>gdb</u>	break to debugger
<u>msg</u>	dump kernel msg buffer to screen
<u>monitor</u>	exit to monitor
<u>reboot</u>	sync and reboot system
<u>halt</u>	sync and halt system
<u>stay</u>	keep this window around
<u>?, help</u>	print this list
<u>{set kernel flag}</u>	set any kernel flag (e.g., debug=2)
<u>{examine kernel flag}</u>	examine any kernel flag (e.g., debug=)

These commands are described in the following sections.

Resume Execution

The **continue** command resumes execution at the point where it left off (that is, the point at which you produced the nonmaskable interrupt).

Break to Debugger

The **gdb** command allows debugging of the system with the GDB debugger using a cooperating computer connected over serial port A.

Dump Kernel Message Buffer to Screen

The **msg** command displays the contents of the kernel message buffer. Messages from the system are recorded in a buffer whose contents can be displayed with this command. This is useful because system messages aren't normally displayed on the screen. (If you want such messages to be displayed, choose the Console menu item in the Workspace Manager's Utilities menu, or open a Terminal window as the Console window. This command allows you to examine those messages even if they haven't been displayed.

Exit to Monitor

The **monitor** command brings you to the ROM monitor prompt (^aNeXT>^o). The ROM monitor commands are described earlier in this chapter. Use the continue (c) command of the ROM monitor to resume running the system.

Sync and Reboot System

The **reboot** command saves all files to the disk and restarts the system with the same boot command used previously.

Sync and Halt System

The **halt** command saves all files to the disk and halts the system.

Keep This Window Around

Normally when you use the NMI **continue** command to resume execution, the NMI window disappears from the screen. If you give the **stay** command before giving the **continue** command, the NMI window will remain on the screen instead.

Print This List

The **help** command (or its equivalent, **?**) can be used to display the list of possible commands and their syntax.

Set or Examine Any Kernel Flag

The command

```
flag=[value]
```

lets you examine or modify internal system flags (most internal system flags are of interest only to system developers).

Example:

```
nmi> debug=
```

```
kernel flags debug = 0x0 The debug flag has the value 0
```

```
nmi> debug=2
```

```
kernel flag = 0x2 Change the value of the debug flag to 2
```

System Test Diagnostics

This section lists error codes that might be returned if a system test diagnostic fails. When the system test fails while the animating boot icon is being displayed, you can display the text window containing the specific system test error message by typing the NMI key sequence (that is, by pressing the backquote key on the keypad while holding down both the left and right Command keys).

FPU Tests

Error Code Meaning

41 FPU data registers are faulty.

42 FPU and CPU intraprocessor data transfer test failed.

43 FPU arithmetic functional test failed.

SCC Channel Tests

Error Code Meaning

51 Port A SCC/DMA transmit interrupt didn't occur within 0.1 second.

52 Port A SCC chip didn't receive the loopback character within 0.1 second.

53 Port A loopback character doesn't match the character sent.

55 Port B Tx buffer not cleared after waiting for 0.1 second.

56 Port B non-DMA transmit not finished within 0.1 second.

57 Port B non-DMA loopback character doesn't match the one sent.

SCSI Channel Tests

Error Code Meaning

61 FIFO count doesn't match the number of characters loaded.

62 FIFO doesn't contain the data that was loaded.

63 FIFO count doesn't match number of characters left in FIFO.

Tests that produce the following error codes are executed only if a SCSI drive is installed and the `p` command's `SCSI test` parameter is enabled.

Error Code Meaning

64 FIFO flush command failed.

65 Cannot load counter with test pattern.

66 Cannot load counter with inversed test pattern.

67 Configuration register failed.

68 53c90 doesn't generate interrupt on illegal command.

Ethernet Channel Tests

Error Code Meaning

71 Couldn't set physical address.

72 Couldn't drain DMA receive buffer.

73 Physical packet not received.

74 Physical packet was garbled.

75 Received a packet with wrong address.

76 Broadcast packet not received.

77 Broadcast packet was garbled.

MO Drive ECC Tests

Error Code Meaning

81 DMA transfer of raw data to ECC failed.

82 DMA read of encoded data from the ECC failed.

83 DMA transfer of corrupted data back to ECC failed.

84 ECC error count is incorrect.

85 DMA transfer of corrected data from ECC failed.

86 ECC corrected data isn't correct.

Real-Time Clock Tests

Error Code Meaning

91 The RTC didn't tick after 0.3 seconds.

Two Bit Graphics Tests

Error Code Meaning

a1 Main memory TBG $A*B$ function failed.

a2 Main memory TBG ceiling($A+B$) function failed.

a3 Main memory TBG $(1-A)*B$ function failed.

a4 Main memory TBG $A+B-A*B$ function failed.

a1 Video memory TBG $A*B$ function failed.

a2 Video memory TBG ceiling($A+B$) function failed.

a3 Video memory TBG $(1-A)*B$ function failed.

a4 Video memory TBG $A+B-A*B$ function failed.

System Timer Tests

Error Code Meaning

c2 The system timer can't be set.

c3 The system timer interrupt doesn't work.

Event Counter Tests

Error Code Meaning

d1 The event counter didn't tick after 2 microseconds.

d2 The event counter isn't accurate.

d3 The event counter isn't ticking smoothly.

Sound-Out Tests

Error Code Meaning

e1 DMA sound-out start failed.

e2 DMA transfer of sound-out data failed.