Netra™ 440 Server Diagnostics and Troubleshooting Guide

Sun Microsystems, Inc.
www.sun.com

Part No. 817-3886-10
April 2004, Revision A

Submit comments about this document at: http://www.sun.com/hwdocs/feedback

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

LA DOCUMENTATION EST FOURNIE "EN L'ÉTAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.

Contents

Figures

Tables

Preface

The *Netra 440 Server Diagnostics and Troubleshooting Guide* is intended to be used by experienced system administrators. It includes descriptive information about the Netra™ 440 server and its diagnostic tools, and specific information about diagnosing and troubleshooting problems with the server.

Before You Read This Book

This book assumes that you are familiar with computer network concepts and terms, and have advanced familiarity with the Solaris™ Operating System (Solaris OS).

To use the information in this document fully, you must have thorough knowledge of the topics discussed in the *Netra 440 Server System Administration Guide* (817-3884-xx).

How This Book Is Organized

The first part of this book is organized a bit differently from others with which you may be familiar. Each chapter contains either conceptual or procedural material, but not both. Turn to the conceptual chapters to get the background information you need to understand the context of the tasks you must perform. Turn to the procedural chapters for quick access to step-by-step instructions with little or no explanatory material.

The chapters in the second part of this book, as well as the Appendix, contain a mixture of procedural and conceptual material.

To help you locate information quickly, the first page of each chapter contains a list that summarizes the topics covered in that chapter. Reference material appears as needed at the end of each chapter.

This book is divided into two parts. Part I covers diagnostic tools.

- Chapter 1, a conceptual chapter, provides an overview of the diagnostic tools available for use with the Netra 440 server.
- Chapter 2, a conceptual chapter, provides detailed information about the uses and capabilities of the various diagnostic tools and explains how they are related to each other.
- Chapter 3, a procedural chapter, provides instructions for isolating failed parts.
- Chapter 4, a procedural chapter, provides instructions for monitoring the system.
- Chapter 5, a procedural chapter, provides instructions for exercising the system.

Part II of this book covers troubleshooting.

- Chapter 6, a conceptual and procedural chapter, explains the troubleshooting options available to you and provides instructions for implementing troubleshooting options.
- Chapter 7, a conceptual and procedural chapter, explains troubleshooting approaches and provides instructions for troubleshooting hardware problems.

Using UNIX Commands

This document might not contain information on basic UNIX® commands and procedures such as shutting down the system, booting the system, and configuring devices. See the following for this information:

- Software documentation that you received with your system
- Solaris OS documentation, which is at

  `http://docs.sun.com`

Shell Prompts

Shell	Prompt
C shell	*machine-name*`%`
C shell superuser	*machine-name*`#`
Bourne shell and Korn shell	`$`
Bourne shell and Korn shell superuser	`#`

Typographic Conventions

Typeface[*]	Meaning	Examples
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `% You have mail.`

`AaBbCc123`	What you type, when contrasted with on-screen computer output	`%` **`su`**
		`Password:`

AaBbCc123	Book titles, new words or terms, words to be emphasized. Replace command-line variables with real names or values.	Read Chapter 6 in the *User's Guide*. These are called *class* options. You *must* be superuser to do this. To delete a file, type `rm` *filename*.

[*] The settings on your browser might differ from these settings.

Related Documentation

Application	Title	Part Number
Late-breaking product information	*Netra 440 Server Product Note*	817-3885-xx
Product description	*Netra 440 Server Product Overview*	817-3881-xx
Installation instructions	*Netra 440 Server Installation Guide*	817-3882-xx
Administration	*Netra 440 Server System Administration Guide*	817-3884-xx
Parts installation and removal	*Netra 440 Server Service Manual*	817-3883-xx
Advanced Lights Out Manager (ALOM) system controller	*Advanced Lights Out Manager User's Guide*	817-xxxx-xx

Accessing Sun Documentation

You can view, print, or purchase a broad selection of Sun documentation, including localized versions, at:

`http://www.sun.com/documentation`

Third-Party Web Sites

Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Sun will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Contacting Sun Technical Support

If you have technical questions about this product that are not answered in this document, go to:

http://www.sun.com/service/contacting

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. You can submit your comments by going to:

http://www.sun.com/hwdocs/feedback

Please include the title and part number of your document with your feedback:

Netra 440 Server Diagnostics and Troubleshooting Guide, part number 817-3886-10

PART I Diagnostics

The five chapters in this part of the *Netra 440 Server Diagnostics and Troubleshooting Guide* introduce the server's hardware-based, firmware-based and software-based diagnostic tools, help you understand how those tools fit together, and tell you how to use the tools to monitor, exercise, and isolate faults in the system.

For information and detailed instructions on how to troubleshoot specific problems with the server, see the chapters in Part II – Troubleshooting.

Part I includes:

CHAPTER 1

Diagnostic Tools Overview

The Netra 440 server and its accompanying software and firmware contain many diagnostic tools and features that can help you:

- *Isolate* problems when there is a failure of a field-replaceable component
- *Monitor* the status of a functioning system
- *Exercise* the system to disclose an intermittent or incipient problem

This chapter introduces the diagnostic tools you can use on the server.

If you want comprehensive background information about diagnostic tools, read this chapter and then read Chapter 2 to find out how the tools fit together.

If you only want instructions for using diagnostic tools, skip the first two chapters and turn to:

- Chapter 3, for part isolating procedures
- Chapter 4, for system monitoring procedures
- Chapter 5, for system exercising procedures

You may also find it helpful to turn to the *Netra 440 Server System Administration Guide* for information about the system console.

A Spectrum of Tools

Sun provides a wide spectrum of diagnostic tools for use with the Netra 440 server. These tools range from the SunVTS™ software, a comprehensive validation test suite, to log files that may contain clues helpful in narrowing down the possible sources of a problem.

The diagnostic tool spectrum also ranges from standalone software packages, to firmware-based power-on self-test (POST), to hardware LEDs that tell you when the power supplies are operating.

Some diagnostic tools enable you to examine many systems from a single console; others do not. Some diagnostic tools stress the system by running tests in parallel, while other tools run sequential tests, enabling the system to continue its normal functions. Some diagnostic tools function on standby power or when the system is offline, while others require the operating system to be up and running.

TABLE 1-1 summarizes the full palette of tools. Most of these tools are discussed in depth in this manual; some are discussed in greater detail in the *Netra 440 Server Administration Guide* (817-3884-xx). Some tools also have their own comprehensive documentation sets. See the Preface for more information.

TABLE 1-1 Summary of Diagnostic Tools

Diagnostic Tool	Type	What It Does	Accessibility and Availability	Remote Capability
Advanced Lights Out Manager (ALOM)	Hardware, software, and firmware	Monitors environmental conditions, generates alerts, performs basic fault isolation, and provides remote console access.	Can function on standby power and when the operating system is not running.	Designed for remote access
LEDs	Hardware	Indicate status of overall system and particular components.	Accessed from system chassis. Available anytime system power is available.	Local, but can be accessed through ALOM
POST	Firmware	Tests core components of system: CPUs, memory, and motherboard I/O bridge integrated circuits.	Can be run on startup, but default is no POST. Available when the operating system is not running.	Local, but can be accessed through ALOM

TABLE 1-1 Summary of Diagnostic Tools *(Continued)*

Diagnostic Tool	Type	What It Does	Accessibility and Availability	Remote Capability
OpenBoot Diagnostics	Firmware	Tests system components, focusing on peripherals and I/O devices.	Can be run automatically at startup, but the default is no diagnostics. Can also be run interactively. Available when the operating system is not running.	Local, but can be accessed through ALOM
OpenBoot commands	Firmware	Display various kinds of system information.	Available when the operating system is not running.	Local, but can be accessed through ALOM
Solaris commands	Software	Display various kinds of system information.	Requires operating system.	Local, and over network
SunVTS	Software	Exercises and stresses the system, running tests in parallel.	Requires operating system. You may need to install SunVTS software separately.	View and control over network

Why are there so many different diagnostic tools?

There are a number of reasons for the lack of a single all-in-one diagnostic test, starting with the complexity of the server.

Consider the bus repeater circuit built into every Netra 440 server. This circuit interconnects all CPUs and high-speed I/O interfaces (see FIGURE 1-1), sensing and adapting its communications depending on how many CPU modules are present. This sophisticated high-speed interconnect represents just one facet of the Netra 440 server's advanced architecture.

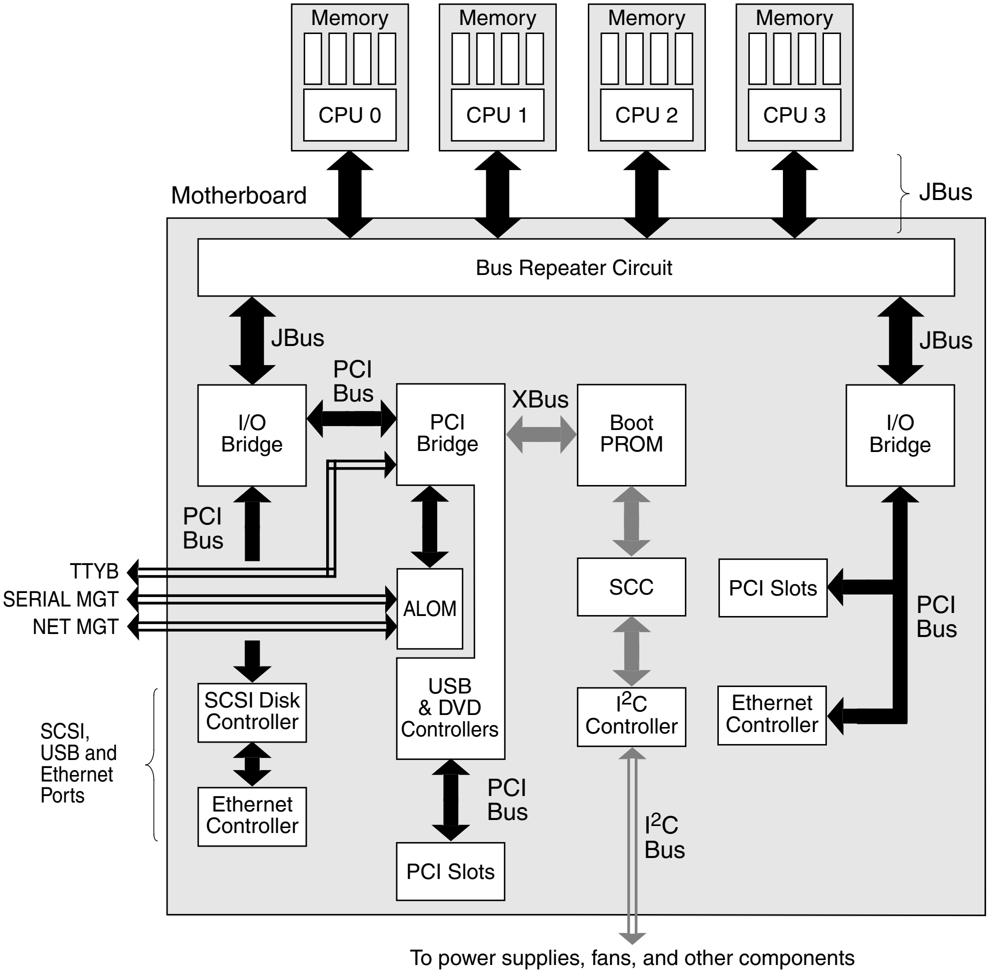


FIGURE 1-1 Simplified Schematic View of a Netra 440 Server

Consider also that some diagnostics must function even when the system fails to boot. Any diagnostic capable of isolating problems when the system fails to boot must be independent of the operating system. But any diagnostic that is independent of the operating system will also be unable to make use of the operating system's considerable resources for getting at the more complex causes of failures.

Another complicating factor is that different sites have differing diagnostic requirements. You may be administering a single computer or a whole data center full of equipment in racks. Alternatively, your systems may be deployed remotely—perhaps in areas that are physically inaccessible.

Finally, consider the different tasks you expect to perform with your diagnostic tools:

- Isolating faults to a specific replaceable hardware component
- Exercising the system to disclose more subtle problems that may or may not be hardware related
- Monitoring the system to catch problems before they become serious enough to cause unplanned downtime

Not every diagnostic tool can be optimized for all these varied tasks.

Instead of one unified diagnostic tool, Sun provides a palette of tools, each of which has its own specific strengths and applications. To best appreciate how each tool fits into the larger picture, it is necessary to have some understanding of what happens when the server starts up, during the so-called *boot process*. This is discussed in the next chapter.

CHAPTER 2

Diagnostics and the Boot Process

This chapter introduces the tools that let you accomplish the goals of isolating faults and monitoring and exercising systems. It also helps you to understand how the various tools fit together.

Topics in this chapter include:

If you only want instructions for using diagnostic tools, skip this chapter and turn to:

- Chapter 3, for part isolating procedures
- Chapter 4, for system monitoring procedures
- Chapter 5, for system exercising procedures

You may also find it helpful to turn to *Netra 440 Server System Administration Guide* for information about the system console.

Diagnostics and the Boot Process

You have probably had the experience of powering on a Sun system and watching as it goes through its boot process. Perhaps you have watched as your console displays messages that look like the following.

```
0>@(#) Netra[TM] 440 POST 4.10.0 2003/04/01 22:28

/export/work/staff/firmware_re/post/post-build
4.10.0/Fiesta/system/integrated  (firmware_re)
0>Hard Powerup RST thru SW
0>CPUs present in system: 0 1 2 3
0>OBP->POST Call with %o0=00000000.01008000.
0>Diag level set to MAX.
0>MFG scrpt mode set to NONE
0>I/O port set to TTYA.
0>
0>Start selftest...
```

It turns out these messages are not quite so inscrutable as they first appear once you understand the boot process. These kinds of messages are discussed later.

It is possible to bypass firmware-based diagnostic tests in order to minimize how long it takes a server to reboot. However, in the following discussion, assume that the system is attempting to boot in *diagnostics mode*, during which the firmware-based tests run. See "Putting the System in Diagnostics Mode" on page 52 for instructions.

The boot process requires several stages, detailed in these sections:

- "System Controller Boot" on page 8
- "OpenBoot Firmware and POST" on page 9
- "OpenBoot Diagnostics Tests" on page 15
- "Operating System" on page 23

System Controller Boot

As soon as you connect the Netra 440 server to an electrical outlet, and before you turn on power to the server, the *system controller* inside the server begins its self-diagnostic and boot cycle. The system controller is incorporated into the Sun™ Remote System Control (ALOM) card installed in the Netra 440 server chassis. Running off standby power, the card begins functioning before the server itself comes up.

The system controller provides access to a number of control and monitoring functions through the ALOM command-line interface. For more information about ALOM, see "Monitoring the System Using Advanced Lights Out Manager" on page 35.

OpenBoot Firmware and POST

Every Netra 440 server includes a chip holding about 2 Mbyte of firmware-based code. This chip is called the *boot PROM*. After you turn on system power, the first thing the system does is execute code that resides in the boot PROM.

This code, which is referred to as the *OpenBoot™ firmware*, is a small-scale operating system unto itself. However, unlike a traditional operating system that can run multiple applications for multiple simultaneous users, OpenBoot firmware runs in single-user mode and is designed solely to configure and boot the system. OpenBoot firmware also initiates firmware-based diagnostics that test the system, thereby ensuring that the hardware is sufficiently "healthy" to run its normal operating environment.

When system power is turned on, the OpenBoot firmware begins running directly out of the boot PROM, since at this stage system memory has not been verified to work properly.

Soon after power is turned on, the system hardware determines that at least one CPU is powered on, and is submitting a bus access request, which indicates that the CPU in question is at least partly functional. This becomes the master CPU, and is responsible for executing OpenBoot firmware instructions.

The OpenBoot firmware's first actions are to check whether to run the *power-on self-test* (POST) diagnostics and other tests. The POST diagnostics constitute a separate chunk of code stored in a different area of the boot PROM (see FIGURE 2-1).

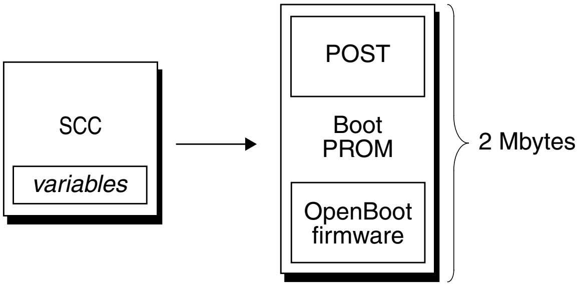


FIGURE 2-1 Boot PROM and SCC

The extent of these power-on self-tests, and whether they are performed at all, is controlled by configuration variables stored in the removable system configuration card (SCC). These *OpenBoot configuration variables* are discussed in “Controlling POST Diagnostics” on page 13.

As soon as POST diagnostics can verify that some subset of system memory is functional, tests are loaded into system memory.

Purpose of POST Diagnostics

The POST diagnostics verify the core functionality of the system. A successful execution of the POST diagnostics does not ensure that there is nothing wrong with the server, but it does ensure that the server can proceed to the next stage of the boot process.

For a Netra 440 server, this means:

- At least one of the CPUs is working.
- At least a subset (512 Mbyte) of system memory is functional.
- Input/output bridges located on the motherboard are functioning.
- The PCI bus is intact—that is, there are no electrical shorts.

It is possible for a system to pass all POST diagnostics and still be unable to boot the operating system. However, you can run POST diagnostics even when a system fails to boot, and these tests are likely to disclose the source of most hardware problems.

POST generally reports errors that are persistent in nature. To catch intermittent problems, consider running a system exercising tool. See “Exercising the System” on page 36.

What POST Diagnostics Do

Each POST diagnostic is a low-level test designed to pinpoint faults in a specific hardware component. For example, individual memory tests called *address bitwalk* and *data bitwalk* ensure that binary *0*s and *1*s can be written on each address and data line. During such a test, the POST may display output similar to this example.

```
1>Data Bitwalk on Slave 3
1>     Test Bank 0.
```

In this example, CPU 1 is the master CPU, as indicated by the prompt `1>`, and it is about to test the memory associated with CPU 3, as indicated by the message `Slave 3`.

The failure of such a test reveals precise information about particular integrated circuits, the memory registers inside them, or the data paths connecting them.

```
1>ERROR: TEST = Data Bitwalk on Slave 3
1>H/W under test = CPU3 B0/D1 J0602 side 1 (Bank 1), CPU Module C3
1>Repair Instructions: Replace items in order listed by 'H/W under test' above
1>MSG = ERROR: miscompare on mem test!
               Address: 00000030.001b0040
               Expected: ffffffff.fffffffe
               Observed: ffffbff.fffffff6
```

In this case, the DIMM labeled J0602, associated with CPU 3, was found to be faulty. For information about the several ways firmware messages identify memory, see "Identifying Memory Modules" on page 39.

What POST Error Messages Tell You

When a specific power-on self-test discloses an error, it reports the following kinds of information about the error:

- The specific test that failed
- The specific integrated circuit or subcomponent that is most likely at fault
- The field-replaceable units (FRUs) most likely to require replacement, in order of likelihood

Here is an excerpt of POST output showing another error message.

CODE EXAMPLE 2-1 POST Error Message

```
1>ERROR: TEST = IO-Bridge unit 0 PCI id    test
1>H/W under test = Motherboard IO-Bridge 0, CPU
1>Repair Instructions: Replace items in order listed by 'H/W under test' above
1>MSG = ERROR: PCI Master Abort Detected for
    TOMATILLO:0, PCI BUS: A, DEVICE NUMBER:2.
    DEVICE NAME: SCSI
1>END_ERROR

1>
1>ERROR: TEST = IO-Bridge unit 0 PCI id    test
1>H/W under test = Motherboard IO-Bridge 0, CPU
1>MSG =
        *** Test Failed!! ***

1>END_ERROR
```

Identifying FRUs

An important feature of POST error messages is the `H/W under test` line (the second line in CODE EXAMPLE 2-1) indicates which FRU or FRUs may be responsible for the error. Note that in CODE EXAMPLE 2-1, two different FRUs are indicated. Using TABLE 2-13 to decode some of the terms, you can see that this POST error was most likely caused by bad integrated circuits (`IO-Bridge`) or electrical pathways on the motherboard. However, the error message also indicates that the master CPU, in this case CPU 1, may be at fault. For information on how Netra 440 CPUs are numbered, see "Identifying CPU/Memory Modules" on page 41.

Though beyond the scope of this manual, it is worth noting that POST error messages provide fault isolation capability beyond the FRU level. In the current example, the `MSG` line located immediately below the `H/W under test` line specifies the particular integrated circuit (`DEVICE NAME: SCSI`) most likely at fault. This level of isolation is most useful at the repair depot.

Why a POST Error Might Implicate Multiple FRUs

Because each test operates at such a low level, the POST diagnostics are often more definite in reporting the minute details of the error, like the numerical values of expected and observed results, than they are about reporting which FRU is responsible. If this seems counterintuitive, consider the block diagram of one data path within a Netra 440 server, shown in FIGURE 2-2.

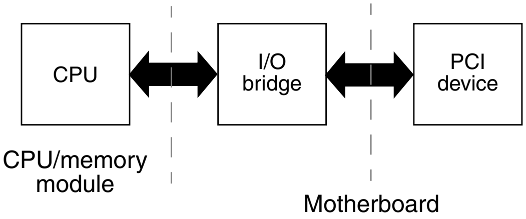


FIGURE 2-2 POST Diagnostic Running Across FRUs

The dashed line in FIGURE 2-2 represents a boundary between FRUs. Suppose a POST diagnostic is running in the CPU in the left part of the diagram. This diagnostic attempts to access registers in a PCI device located in the right side of the diagram.

If this access fails, there could be a fault in the PCI device, or, less likely, in one of the data paths or components leading to that PCI device. The POST diagnostic can tell you only that the test failed, but not *why*. So, though the POST diagnostic may present very precise data about the nature of the test failure, potentially several different FRUs could be implicated.

Controlling POST Diagnostics

You control POST diagnostics (and other aspects of the boot process) by setting OpenBoot configuration variables in the system configuration card. Changes to OpenBoot configuration variables generally take effect only after the server is reset.

TABLE 2-1 lists the most important and useful of these variables, which are more fully documented in the *OpenBoot Command Reference Manual*. You can find instructions for changing OpenBoot configuration variables in "Viewing and Setting OpenBoot Configuration Variables" on page 50.

TABLE 2-1 OpenBoot Configuration Variables

OpenBoot Configuration Variable	Description and Keywords
`auto-boot?`	Determines whether the operating system automatically starts up. Default is `true`. • `true`—Operating system automatically starts once OpenBoot firmware completes initialization. • `false`—System remains at `ok` prompt until you type `boot`.
`diag-level`	Determines the level or type of diagnostics executed. Default is . • `off`—No testing. • `min`—Only basic tests are run. • `max`—More extensive tests may be run, depending on the device. Memory is especially thoroughly checked.
`diag-script`	Determines which devices are tested by OpenBoot Diagnostics. Default is `none`. • `none`—No devices are tested. • `normal`—On-board (motherboard-based) devices that have self-tests are tested. • `all`—All devices that have self-tests are tested.
`diag-switch?`	 • `true`—if `post-trigger` and `obdiag-trigger` conditions, respectively, are satisfied. Causes system to boot using `diag-device` and `diag-file` parameters. `false`— , even if `post-trigger` and `obdiag-trigger` conditions are satisfied. Causes system to boot using `boot-device` and `boot-file` parameters.**NOTE**: You can put the system in diagnostics mode either by setting this variable to `true` or by setting the system control rotary switch to the Diagnostics position. For details, see "Putting the System in Diagnostics Mode" on page 52.

TABLE 2-1 OpenBoot Configuration Variables *(Continued)*

OpenBoot Configuration Variable	Description and Keywords
post-trigger obdiag-trigger	Specifies the class of reset event that causes POST diagnostics or OpenBoot Diagnostics tests to run. These variables can accept single keywords as well as combinations of the first three keywords separated by spaces. For details, see "Viewing and Setting OpenBoot Configuration Variables" on page 50. • error-reset—A reset caused by certain nonrecoverable hardware error conditions. In general, an error reset occurs when a hardware problem corrupts system state data and the machine becomes "confused." Examples include CPU and system watchdog resets, fatal errors, and certain CPU reset events (default). • power-on-reset—A reset caused by pressing the Power button (default). • user-reset—A reset initiated by the user or the operating system. Examples of user resets include the OpenBoot boot and reset-all commands, as well as the Solaris reboot command. • all-resets—Any kind of system reset. • none—No POST diagnostics or OpenBoot Diagnostics tests run.
input-device	Selects where system console input is taken from. Default is ttya. • ttya—From serial and network management ports. • ttyb—From built-in serial port B.* • keyboard—From attached keyboard that is part of a local graphics monitor.*
output-device	Selects where diagnostic and other system console output is displayed. Default is ttya. • ttya—To serial and network management ports. • ttyb—To built-in serial port B.* • screen—To attached screen that is part of a local graphics monitor.*

* POST messages cannot be displayed on a local graphics monitor. They are sent to ttya even when output-device is set to screen. Likewise, POST can accept input only from ttya.

Note – These variables affect OpenBoot Diagnostics tests as well as POST diagnostics.

Diagnostics: Reliability versus Availability

The OpenBoot configuration variables described in TABLE 2-1 let you control not only how diagnostic tests proceed, but also what triggers them.

Bypassing diagnostic tests can create a situation where a server with faulty hardware gets locked into a cycle of repeated booting and crashing. Depending on the type of problem, the cycle may repeat intermittently. Because diagnostic tests are never invoked, the crashes may occur without leaving behind any log entries or meaningful console messages.

The section "Putting the System in Diagnostics Mode" on page 52 provides instructions for ensuring that your server runs diagnostics when starting up. The section "Bypassing Firmware Diagnostics" on page 54 explains how to disable firmware diagnostics.

Temporarily Bypassing Diagnostics

Even if you set up the server to run diagnostic tests automatically on reboot, it is still possible to bypass diagnostic tests for a single boot cycle. This can be useful in cases where you are reconfiguring the server, or on those rare occasions when POST or OpenBoot Diagnostics tests themselves stall or "hang," leaving the server unable to boot and in an unusable state. These "hangs" most commonly result from firmware corruption of some sort, especially of having flashed an incompatible firmware image into the server's PROMs.

If you do find yourself needing to skip diagnostic tests for a single boot cycle, the ALOM system controller provides a convenient way to do this. See "Bypassing Diagnostics Temporarily" on page 55 for instructions.

Maximizing Reliability

By default, diagnostics do not run following a user- or operating system-initiated reset. This means the system does not run diagnostics in the event of an operating system panic. To ensure the maximum reliability, especially for automatic system recovery (ASR), you can configure the system to run its firmware-based diagnostic tests following all resets. For instructions, see "Maximizing Diagnostic Testing" on page 56.

OpenBoot Diagnostics Tests

Once POST diagnostics have finished running, POST marks the status of any faulty device as "FAILED," and returns control to OpenBoot firmware.

OpenBoot firmware compiles a hierarchical "census" of all devices in the system. This census is called a *device tree*. Though different for every system configuration, the device tree generally includes both built-in system components and optional PCI bus devices. The device tree does *not* include any components marked as "FAILED" by POST diagnostics.

Following the successful execution of POST diagnostics, the OpenBoot firmware proceeds to run OpenBoot Diagnostics tests. Like the POST diagnostics, OpenBoot Diagnostics code is firmware-based and resides in the boot PROM.

Purpose of OpenBoot Diagnostics Tests

OpenBoot Diagnostics tests focus on system I/O and peripheral devices. Any device in the device tree, regardless of manufacturer, that includes an IEEE 1275-compatible self-test is included in the suite of OpenBoot Diagnostics tests. On a Netra 440 server, OpenBoot Diagnostics examine the following system components:

- I/O interfaces; including USB and serial ports, SCSI and IDE controllers, and Ethernet interfaces
- ALOM system controller card
- Keyboard, mouse, and video (when present)
- Inter-Integrated Circuit (I^2C) bus components; including thermal and other kinds of sensors located on the motherboard, CPU/memory modules, DIMMs, power supply, and SCSI backplane
- Any PCI option card with an IEEE 1275-compatible built-in self-test

The OpenBoot Diagnostics tests run automatically through a script when you start up the system in diagnostics mode. However, you can also run OpenBoot Diagnostics tests manually, as explained in the next section.

Like POST diagnostics, OpenBoot Diagnostics tests catch persistent errors. To disclose intermittent problems, consider running a system exercising tool. See "Exercising the System" on page 36.

Controlling OpenBoot Diagnostics Tests

When you restart the system, you can run OpenBoot Diagnostics tests either interactively from a test menu, or by entering commands directly from the `ok` prompt.

Note – You cannot reliably run OpenBoot Diagnostics tests following an operating system halt, since the halt leaves system memory in an unpredictable state. Best practice is to reset the system before running these tests.

Most of the same OpenBoot configuration variables you use to control POST (see TABLE 2-1) also affect OpenBoot Diagnostics tests. Notably, you can determine OpenBoot Diagnostics testing level—or suppress testing entirely—by appropriately setting the `diag-level` variable.

In addition, the OpenBoot Diagnostics tests use a special variable called `test-args` that enables you to customize how the tests operate. By default, `test-args` is set to contain an empty string. However, you can set `test-args` to one or more of the reserved keywords, each of which has a different effect on OpenBoot Diagnostics tests. TABLE 2-2 lists the available keywords.

TABLE 2-2 Keywords for the `test-args` OpenBoot Configuration Variable

Keyword	What It Does
`bist`	Invokes built-in self-test (BIST) on external and peripheral devices
`debug`	Displays all debug messages
`iopath`	Verifies bus and interconnect integrity
`loopback`	Exercises external loopback path for the device
`media`	Verifies external and peripheral device media accessibility
`restore`	Attempts to restore original state of the device if the previous execution of the test failed
`silent`	Displays only errors rather than the status of each test
`subtests`	Displays main test and each subtest that is called
`verbose`	Displays detailed messages of status of all tests
`callers=`*N*	Displays backtrace of *N* callers when an error occurs • `callers=0` — Displays backtrace of all callers before the error
`errors=`*N*	Continues executing the test until *N* errors are encountered • `errors=0` — Displays all error reports without terminating testing

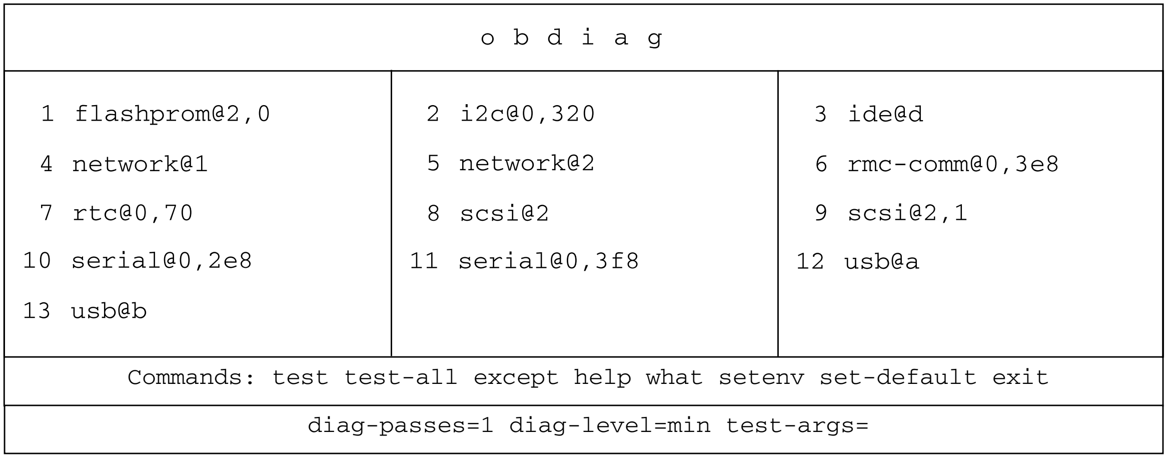
If you want to make multiple customizations to the OpenBoot Diagnostics testing, you can set `test-args` to a comma-separated list of keywords, as in this example:

```
ok setenv test-args debug,loopback,media
```

From the OpenBoot Diagnostics Test Menu

It is easiest to run OpenBoot Diagnostics tests interactively from a menu. You access the menu by typing `obdiag` at the `ok` prompt. See "Isolating Faults Using Interactive OpenBoot Diagnostics Tests" on page 62 for full instructions.

The `obdiag>` prompt and the OpenBoot Diagnostics interactive menu (FIGURE 2-3) appear. Only the devices detected by OpenBoot firmware appear in this menu. For a brief explanation of each OpenBoot Diagnostics test, see TABLE 2-10 in "OpenBoot Diagnostics Test Descriptions" on page 43.



```
                              obdiag

  1 flashprom@2,0        2 i2c@0,320          3 ide@d
  4 network@1            5 network@2          6 rmc-comm@0,3e8
  7 rtc@0,70             8 scsi@2             9 scsi@2,1
 10 serial@0,2e8        11 serial@0,3f8      12 usb@a
 13 usb@b

      Commands: test test-all except help what setenv set-default exit

                  diag-passes=1 diag-level=min test-args=
```

FIGURE 2-3 OpenBoot Diagnostics Interactive Test Menu

Interactive OpenBoot Diagnostics Commands

You run individual OpenBoot Diagnostics tests from the `obdiag>` prompt by typing:

```
obdiag> test n
```

where *n* represents the number associated with a particular menu item.

Note – You cannot reliably run OpenBoot Diagnostics commands following an operating system halt, since the halt leaves system memory in an unpredictable state. Best practice is to reset the system before running these commands.

There are several other commands available to you from the `obdiag>` prompt. For descriptions of these commands, see TABLE 2-11 in "OpenBoot Diagnostics Test Descriptions" on page 43.

You can obtain a summary of this same information by typing `help` at the `obdiag>` prompt.

From the `ok` *Prompt: The* `test` *and* `test-all` *Commands*

You can also run OpenBoot Diagnostics tests directly from the `ok` prompt. To do this, type the `test` command, followed by the full hardware path of the device (or set of devices) to be tested. For example:

```
ok test /pci@1c,600000/scsi@2,1
```

Note – Knowing how to construct an appropriate hardware device path requires precise knowledge of the hardware architecture of the Netra 440 server. If you lack this knowledge, it may help to use the OpenBoot `show-devs` command (see "`show-devs` Command" on page 23), which displays a list of all configured devices.

To customize an individual test, you can use `test-args` as follows:

```
ok test /pci@1e,600000/usb@b:test-args={verbose,subtests}
```

This affects only the current test without changing the value of the `test-args` OpenBoot configuration variable.

You can test all the devices in the device tree with the `test-all` command:

```
ok test-all
```

If you specify a path argument to `test-all`, then only the specified device and its children are tested. The following example shows the command to test the USB bus and all devices with self-tests that are connected to the USB bus:

```
ok test-all /pci@1f,700000
```

Note – You cannot reliably run OpenBoot Diagnostics commands following an operating system halt, since the halt leaves system memory in an unpredictable state. Best practice is to reset the system before running these commands.

What OpenBoot Diagnostics Error Messages Tell You

OpenBoot Diagnostics error messages are reported in a tabular format that contains a short summary of the problem, the hardware device affected, the subtest that failed, and other diagnostic information. CODE EXAMPLE 2-2 displays a sample OpenBoot Diagnostics error message, one that suggests a failure of the IDE controller.

CODE EXAMPLE 2-2 OpenBoot Diagnostics Error Message

```
Testing /pci@1e,600000/ide@d

    ERROR   : IDE device did not reset, busy bit not set
    DEVICE  : /pci@1e,600000/ide@d
    DEVICE  : /pci@1e,600000/ide@d
    ex MACHINE : Netra 440
    SERIAL# : 51994289
    DATE    : 10/17/2002 20:17:43  GMT
    CONTR0LS: diag-level=min test-args=

Error: /pci@1e,600000/ide@d selftest failed, return code = 1
Selftest at /pci@1e,600000/ide@d (errors=1) .......................... failed
```

I^{2}C Bus Device Tests

The `i2c@0,320` OpenBoot Diagnostics test examines and reports on environmental monitoring and control devices connected to the Netra 440 server's Inter-Integrated Circuit (I^{2}C) bus.

Error and status messages from the `i2c@0,320` OpenBoot Diagnostics test include the hardware addresses of I^{2}C bus devices.

```
Testing /pci@1e,600000/isa@7/i2c@0,320/dimm-spd@0,b6
```

The I^{2}C device address is given at the very end of the hardware path. In this example, the address is `0,b6`, which indicates a device located at hexadecimal address b6 on segment 0 of the I^{2}C bus.

To decode this device address, see "Decoding I^{2}C Diagnostic Test Messages" on page 44. Using TABLE 2-12, you can see that `dimm-spd@0,b6` corresponds to DIMM 0 on CPU/memory module 0. If the `i2c@0,320` test were to report an error against `dimm-spd@0,b6`, you would need to replace this DIMM.

Other OpenBoot Commands

Beyond the formal firmware-based diagnostic tools, there are a few commands you can invoke from the `ok` prompt. These OpenBoot commands display information that can help you assess the condition of a Netra 440 server. These include the following:

- `printenv` command
- `probe-scsi` and `probe-scsi-all` commands
- `probe-ide` command
- `show-devs` command

The following sections describe the information these commands give you. For instructions on using these commands, turn to "Using OpenBoot Information Commands" on page 83, or look up the appropriate man page.

`printenv` *Command*

The `printenv` command displays the OpenBoot configuration variables. The display includes the current values for these variables as well as the default values. For details, see "Viewing and Setting OpenBoot Configuration Variables" on page 50.

For a list of some important OpenBoot configuration variables, see TABLE 2-1.

`probe-scsi` *and* `probe-scsi-all` *Commands*

The `probe-scsi` and `probe-scsi-all` commands diagnose problems with attached and internal SCSI devices.

Caution – If you used the `halt` command or the L1-A (Stop-A) key sequence to reach the `ok` prompt, then issuing the `probe-scsi` or `probe-scsi-all` command can hang the system.

The `probe-scsi` command communicates with all SCSI devices connected to on-board SCSI controllers. The `probe-scsi-all` command additionally accesses devices connected to any host adapters installed in PCI slots.

For any SCSI device that is connected and active, the `probe-scsi` and `probe-scsi-all` commands display its target and unit numbers, and a device description that includes type and manufacturer.

The following is sample output from the `probe-scsi` command.

CODE EXAMPLE 2-3 `probe-scsi` Command Output

```
ok probe-scsi
Target 0
  Unit 0   Disk     FUJITSU MAN3367M SUN36G 1502    71132959 Blocks, 34732 MB
Target 1
  Unit 0   Disk     FUJITSU MAN3367M SUN36G 1502    71132959 Blocks, 34732 MB
```

The following is sample output from the `probe-scsi-all` command.

CODE EXAMPLE 2-4 `probe-scsi-all` Command Output

```
ok probe-scsi-all
/pci@1f,700000/scsi@2,1

/pci@1f,700000/scsi@2
Target 0
  Unit 0   Disk     FUJITSU MAN3367M SUN36G 1502    71132959 Blocks, 34732 MB
Target 1
  Unit 0   Disk     FUJITSU MAN3367M SUN36G 1502    71132959 Blocks, 34732 MB
```

`probe-ide` *Command*

The `probe-ide` command communicates with all Integrated Drive Electronics (IDE) devices connected to the IDE bus. This is the internal system bus for media devices such as the DVD-ROM drive.

Caution – If you used the `halt` command or the L1-A (Stop-A) key sequence to reach the `ok` prompt, then issuing the `probe-ide` command can hang the system.

The following is sample output from the `probe-ide` command.

CODE EXAMPLE 2-5 `probe-ide` Command Output

```
ok probe-ide
Device 0  ( Primary Master )
       Removable ATAPI Model: TOSHIBA DVD-ROM SD-C2512

  Device 1  ( Primary Slave )
         Not Present
```

show-devs *Command*

The show-devs command lists the hardware device paths for each device in the firmware device tree. CODE EXAMPLE 2-6 shows some sample output (edited for brevity).

CODE EXAMPLE 2-6 show-devs Command Output

```
ok show-devs
/i2c@1f,464000
/pci@1f,700000
/ppm@1e,0
/pci@1e,600000
/pci@1d,700000
/ppm@1c,0
/pci@1c,600000
/memory-controller@2,0
/SUNW,UltraSPARC-IIIi@2,0
/virtual-memory
/memory@m0,10
/aliases
/options
/openprom
/packages
/i2c@1f,464000/idprom@0,50
```

Operating System

If a system passes OpenBoot Diagnostics tests, it normally attempts to boot its multiuser operating environment. For most Sun systems, this means the Solaris OS. Once the server is running in multiuser mode, you have recourse to software-based diagnostic tools, like SunVTS and Sun™ Management Center software. These tools can help you with more advanced monitoring, exercising, and fault isolating capabilities.

Note – If you set the auto-boot? OpenBoot configuration variable to false, the operating environment does *not* boot following completion of the firmware-based tests.

In addition to the formal tools that run on top of Solaris OS software, there are other resources that you can use when assessing or monitoring the condition of a Netra 440 server. These resources include the following:

- Error and system message log files
- Solaris system information commands

Error and System Message Log Files

Error and other system messages are saved in the file /var/adm/messages. Messages are logged to this file from many sources, including the operating system, the environmental control subsystem, and various software applications.

In the case of Solaris OS software, the syslogd daemon and its configuration file (/etc/syslogd.conf) control how error messages are handled.

For information about /var/adm/messages and other sources of system information, refer to "How to Customize System Message Logging" in the *System Administration Guide: Advanced Administration*, which is part of the Solaris System Administration Collection.

Solaris System Information Commands

Some Solaris commands display data that you can use when assessing the condition of a Netra 440 server. These commands include the following:

- prtconf command
- prtdiag command
- prtfru command
- psrinfo command
- showrev command

The following sections describe the information these commands give you. For instructions on using these commands, turn to "Using Solaris System Information Commands" on page 82, or look up the appropriate man page.

prtconf *Command*

The prtconf command displays the Solaris device tree. This tree includes all the devices probed by OpenBoot firmware, as well as additional devices, like individual disks, that only the operating environment software "knows" about. The output of prtconf also includes the total amount of system memory. CODE EXAMPLE 2-7 shows an excerpt of prtconf output (edited for brevity).

CODE EXAMPLE 2-7 `prtconf` Command Output

```
System Configuration:  Sun Microsystems  sun4u
Memory size: 16384 Megabytes
System Peripherals (Software Nodes):

SUNW,Netra-440
    packages (driver not attached)
        SUNW,builtin-drivers (driver not attached)
        deblocker (driver not attached)
        disk-label (driver not attached)

[...]

    pci, instance #1
        isa, instance #0
            flashprom (driver not attached)
            rtc (driver not attached)
            i2c, instance #0
                i2c-bridge (driver not attached)
                i2c-bridge (driver not attached)
                temperature (driver not attached)

[...]
```

The `prtconf` command's `-p` option produces output similar to the OpenBoot `show-devs` command (see "`show-devs` Command" on page 23). This output lists only those devices compiled by the system firmware.

`prtdiag` *Command*

The `prtdiag` command displays a table of diagnostic information that summarizes the status of system components.

The display format used by the `prtdiag` command can vary depending on what version of the Solaris OS is running on your system. Following are several excerpts of the output produced by `prtdiag` on a "healthy" Netra 440 server running Solaris 8 software.

CODE EXAMPLE 2-8 `prtdiag` CPU and I/O Output

```
System Configuration: Sun Microsystems  sun4u Netra 440
System clock frequency: 183 MHZ
Memory size: 16GB

==================================== CPUs ====================================
               E$          CPU                  CPU
CPU  Freq      Size        Implementation       Mask    Status      Location
---  --------  ----------  -------------------  -----   ------      --------
  0  1281 MHz  1MB         SUNW,UltraSPARC-IIIi  2.3    online       -
  1  1281 MHz  1MB         SUNW,UltraSPARC-IIIi  2.3    online       -
  2  1281 MHz  1MB         SUNW,UltraSPARC-IIIi  2.3    online       -
  3  1281 MHz  1MB         SUNW,UltraSPARC-IIIi  2.3    online       -

================================= IO Devices =================================
Bus   Freq      Slot +  Name +
Type  MHz       Status  Path                          Model
----  ----  ----------  ----------------------------  --------------------
pci    66           MB  pci108e,abba (network)        SUNW,pci-ce
                  okay  /pci@1c,600000/network@2

pci    33           MB  isa/su (serial)
                  okay  /pci@1e,600000/isa@7/serial@0,3f8

pci    33           MB  isa/su (serial)
                  okay  /pci@1e,600000/isa@7/serial@0,2e8

pci    66           MB  pci108e,abba (network)        SUNW,pci-ce
                  okay  /pci@1f,700000/network@1

pci    66           MB  scsi-pci1000,30 (scsi-2)      LSI,1030
                  okay  /pci@1f,700000/scsi@2
```

The `prtdiag` command produces a great deal of output about the system memory configuration. Another excerpt follows.

CODE EXAMPLE 2-9 `prtdiag` Memory Configuration Output

```
============================ Memory Configuration ============================
Segment Table:
-----------------------------------------------------------------------
Base Address       Size       Interleave Factor  Contains
-----------------------------------------------------------------------
0x0                4GB               16          BankIDs 0,1,2,3, ... ,15
0x1000000000       4GB               16          BankIDs 16,17,18, ... ,31
0x2000000000       4GB               16          BankIDs 32,33,34, ... ,47
0x3000000000       4GB               2           BankIDs 48,49

Bank Table:
-----------------------------------------------------------
        Physical       Location
ID      ControllerID   GroupID   Size       Interleave Way
-----------------------------------------------------------
0        0             0         256MB      0,1,2,3, ... ,15
1        0             0         256MB

[...]

48       3             0         2GB        0,1
49       3             0         2GB

Memory Module Groups:
--------------------------------------------------
ControllerID   GroupID  Labels         Status
--------------------------------------------------
0              0        C0/P0/B0/D0
0              0        C0/P0/B0/D1

[...]

3              0        C3/P0/B0/D1
```

In addition to the preceding information, `prtdiag` with the verbose option (`-v`) also reports on front panel status, disk status, fan status, power supplies, hardware revisions, and system temperatures.

CODE EXAMPLE 2-10 `prtdiag` Verbose Output

```
Temperature sensors:
---------------------------------------------------------------
Location   Sensor      Temperature  Lo LoWarn HiWarn  Hi Status
---------------------------------------------------------------
SCSIBP     T_AMB         26C     -11C     0C    65C    75C okay
C0/P0      T_CORE        55C     -10C     0C    97C   102C okay
```

In the event of an overtemperature condition, `prtdiag` reports `warning` or `failed` in the Status column.

CODE EXAMPLE 2-11 `prtdiag` Overtemperature Indication Output

```
Temperature sensors:
---------------------------------------------------------------
Location   Sensor      Temperature  Lo LoWarn HiWarn  Hi Status
---------------------------------------------------------------
SCSIBP     T_AMB         26C     -11C     0C   65C   75C okay
C0/P0      T_CORE        99C     -10C     0C   97C  102C failed
```

Similarly, if there is a failure of a particular component, `prtdiag` reports a fault in the appropriate Status column.

CODE EXAMPLE 2-12 `prtdiag` Fault Indication Output

```
Fan Status:
---------------------------------------
Location       Sensor          Status
---------------------------------------
FT1/F0         F0              failed (0 rpm)
```

Here is an example of how the `prtdiag` command displays the status of system LEDs.

CODE EXAMPLE 2-13 `prtdiag` LED Status Display

```
Led State:
--------------------------------------------------
Location   Led                   State       Color
--------------------------------------------------
MB         ACT                   on          green
MB         SERVICE               on          amber
MB         LOCATE                off         white
PS0        POK                   off         green
PS0        STBY                  off         green
```

`prtfru` *Command*

The Netra 440 server maintains a hierarchical list of all field-replaceable units (FRUs) in the system, as well as specific information about various FRUs.

The prtfru command can display this hierarchical list, as well as data contained in the serial electrically-erasable programmable read-only memory (SEEPROM) devices located on many FRUs. CODE EXAMPLE 2-14 shows an excerpt of a hierarchical list of FRUs generated by the prtfru command with the -l option.

CODE EXAMPLE 2-14 prtfru -l Command Output

```
/frutree
/frutree/chassis (fru)
/frutree/chassis/SYS?Label=SYS
/frutree/chassis/SYS?Label=SYS/led-location (fru)
/frutree/chassis/SYS?Label=SYS/key-location (fru)
/frutree/chassis/SYS?Label=SYS/key-location/SYSCTRL?Label=SYSCTRL
/frutree/chassis/SC?Label=SC
[...]
/frutree/chassis/HDD0?Label=HDD0
/frutree/chassis/HDD0?Label=HDD0/disk (fru)
/frutree/chassis/HDD1?Label=HDD1
/frutree/chassis/HDD1?Label=HDD1/disk (fru)
/frutree/chassis/HDD2?Label=HDD2
/frutree/chassis/HDD2?Label=HDD2/disk (fru)
/frutree/chassis/HDD3?Label=HDD3
/frutree/chassis/HDD3?Label=HDD3/disk (fru)
/frutree/chassis/DVD?Label=DVD
/frutree/chassis/DVD?Label=DVD/cdrom (fru)
/frutree/chassis/SCC?Label=SCC
/frutree/chassis/SCC?Label=SCC/scc (fru)
/frutree/chassis/ALARM?Label=ALARM
/frutree/chassis/ALARM?Label=ALARM/alarm (container)
[...]
/frutree/chassis/PDB?Label=PDB
/frutree/chassis/PDB?Label=PDB/pdb (container)
```

CODE EXAMPLE 2-15 shows an excerpt of SEEPROM data generated by the prtfru command with the -c option.

CODE EXAMPLE 2-15 prtfru -c Command Output

```
/frutree/chassis/SC?Label=SC/system-controller (container)
   SEGMENT: SD
      /ManR
      /ManR/UNIX_Timestamp32: Wed Dec 31 19:00:00 EST 1969
      /ManR/Fru_Description: ASSY,ALOM Card
      /ManR/Manufacture_Loc:
      /ManR/Sun_Part_No: 5016346
      /ManR/Sun_Serial_No:
      /ManR/Vendor_Name: NO JEDEC CODE FOR THIS VENDOR
      /ManR/Initial_HW_Dash_Level: 03
      /ManR/Initial_HW_Rev_Level:
```

CODE EXAMPLE 2-15 `prtfru -c` Command Output *(Continued)*

```
      /ManR/Fru_Shortname: ALOM_Card
      /SpecPartNo: 885-0084-05
/frutree/chassis/MB?Label=MB/system-board (container)
   SEGMENT: SD
      /ManR
      /ManR/UNIX_Timestamp32: Mon Nov  4 15:35:24 EST 2002
      /ManR/Fru_Description: ASSY,A42,MOTHERBOARD
      /ManR/Manufacture_Loc: Celestica,Toronto,Ontario
      /ManR/Sun_Part_No: 5016344
      /ManR/Sun_Serial_No: 000001
      /ManR/Vendor_Name: Celestica
      /ManR/Initial_HW_Dash_Level: 03
      /ManR/Initial_HW_Rev_Level: 06
      /ManR/Fru_Shortname: A42_MB
      /SpecPartNo: 885-0060-02
```

The `prtfru` command displays varied data depending on the type of FRU. In general, this information includes:

- FRU description
- Manufacturer name and location
- Part number and serial number
- Hardware revision levels

Information about the following Netra 440 server FRUs is displayed by the `prtfru` command:

- ALOM system controller card
- CPU modules
- DIMMs
- Motherboard
- SCSI backplane
- Power supplies

Similar information is provided by the ALOM system controller `showfru` command. For more information about `showfru` and other ALOM commands, see "Monitoring the System Using Sun Advanced Lights Out Manager" on page 68.

`psrinfo` *Command*

The `psrinfo` command displays the date and time each CPU came online. With the verbose option (`-v`), the command displays additional information about the CPUs, including their clock speed. The following is sample output from the `psrinfo` command with the `-v` option.

CODE EXAMPLE 2-16 `psrinfo -v` Command Output

```
Status of processor 0 as of: 04/11/03 12:03:45
  Processor has been on-line since 04/11/03 10:53:03.
  The sparcv9 processor operates at 1280 MHz,
        and has a sparcv9 floating point processor.
Status of processor 1 as of: 04/11/03 12:03:45
  Processor has been on-line since 04/11/03 10:53:05.
  The sparcv9 processor operates at 1280 MHz,
        and has a sparcv9 floating point processor.
```

`showrev` *Command*

The `showrev` command displays revision information for the current hardware and software. CODE EXAMPLE 2-17 shows sample output of the `showrev` command.

CODE EXAMPLE 2-17 `showrev` Command Output

```
Hostname: wgs94-111
Hostid: 83195f01
Release: 5.8
Kernel architecture: sun4u
Application architecture: sparc
Hardware provider: Sun_Microsystems
Domain: Ecd.East.Sun.COM
Kernel version: SunOS 5.8 system28_11:12/03/02 2002
    SunOS Internal Development: root 12/03/02 [system28-gate]
```

When used with the `-p` option, this command displays installed patches. CODE EXAMPLE 2-18 shows a partial sample output from the `showrev` command with the `-p` option.

CODE EXAMPLE 2-18 `showrev -p` Command Output

```
Patch: 112663-01 Obsoletes:  Requires: 108652-44 Incompatibles:  Packages:
SUNWxwplt
Patch: 111382-01 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWxwplt
Patch: 111626-02 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWolrte,
SUNWolslb
Patch: 111741-02 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWxwmod,
SUNWxwmox
Patch: 111844-02 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWxwopt
Patch: 112781-01 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWxwopt
Patch: 108714-07 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWdtbas,
SUNWdtbax
```

Tools and the Boot Process: A Summary

Different diagnostic tools are available to you at different stages of the boot process. TABLE 2-3 summarizes what tools are available to you and when they are available.

TABLE 2-3 Diagnostic Tool Availability

	Available Diagnostic Tools		
Stage	**Fault Isolation**	**System Monitoring**	**System Exercising**
Before the operating system starts	- LEDs - POST - OpenBoot Diagnostics	- ALOM - OpenBoot commands	–none–
After the operating system starts	- LEDs	- ALOM - Solaris info commands	- SunVTS - Hardware Diagnostic Suite
When the system is turned off but standby power is available	–none–	- ALOM	–none–

Isolating Faults in the System

Each of the tools available for fault isolation discloses faults in different field-replaceable units (FRUs). The row headings along the left of TABLE 2-4 list the FRUs in a Netra 440 server. The available diagnostic tools are shown in column headings across the top. A check mark in this table indicates that a fault in a particular FRU can be isolated by a particular diagnostic.

TABLE 2-4 FRU Coverage of Fault–Isolating Tools

		LEDs			
FRU	**ALOM**	**Enclosure**	**On FRU**	**OpenBoot Diags**	**POST**
ALOM system controller card	✓		✓	✓	
Connector board assembly	No coverage. See TABLE 2-5 for fault isolation hints.				
CPU/memory module	✓	✓			✓
DIMMs		✓			✓
Hard drive	✓	✓	✓	✓	
DVD drive			✓	✓	

TABLE 2-4 FRU Coverage of Fault–Isolating Tools *(Continued)*

FRU	ALOM	LEDs		OpenBoot Diags	POST
		Enclosure	On FRU		
Fan tray 3	✓	✓			
Fan trays 0-2	✓	✓			
Motherboard	✓	✓		✓	✓
Power supply	✓	✓	✓		
SCSI backplane	No coverage. See TABLE 2-5 for fault isolation hints.				
System configuration card reader	No coverage. See TABLE 2-5 for fault isolation hints.				
System configuration card	No coverage. See TABLE 2-5 for fault isolation hints.				

In addition to the FRUs listed in TABLE 2-4, there are several minor replaceable system components—mostly cables—that cannot directly be isolated by any system diagnostic. For the most part, you determine when these components are faulty by eliminating other possibilities. Some of these FRUs are listed in TABLE 2-5, along with hints on how to discern problems with them.

TABLE 2-5 FRUs Not Directly Isolated by Fault–Isolating Tools

FRU	Diagnostic Hints
Connector board assembly	This is difficult to distinguish from other problems with similar symptoms. The firmware generates many error messages about being unable to access OpenBoot configuration variables, for example: `Could not read diag-level from NVRAM!` ALOM shows the front panel Service Required indicator is lit.
Connector board power cable	If ALOM is able to read the system rotary switch position, but reports that none of the fans are spinning, you should suspect that this cable is loose or defective.
DVD drive cable	If OpenBoot Diagnostics tests indicate a problem with the DVD drive, but replacing the drive does not fix the problem, you should suspect (primarily) that this cable is either defective or improperly connected, or (secondarily) that there is a problem with the motherboard.
SCSI backplane	Though not an exhaustive diagnostic, some SunVTS tests (`i2c2test` and `disktest`) exercise certain SCSI backplane paths. You can also monitor the backplane's ambient temperature using the ALOM system controller `showenvironment` command (see "Monitoring the System Using Sun Advanced Lights Out Manager" on page 68).

TABLE 2-5 FRUs Not Directly Isolated by Fault–Isolating Tools *(Continued)*

FRU	Diagnostic Hints
SCSI data cable	This is difficult to distinguish from problems with similar symptoms. The firmware generates many error messages about being unable to access OpenBoot configuration variables, for example: `Could not read diag-level from NVRAM!` ALOM shows the front panel Service Required indicator is lit.
System configuration card reader –and– System configuration card reader cable	If the system control rotary switch and On/Standby button appear unresponsive, and if the power supplies are known to be good, you should suspect the SCC reader and its cable. To test these components, access ALOM, issue the `resetsc` command, log in again to ALOM, and remove the system controller card. If an alert message appears ("`SCC card has been removed`"), it means the card reader is functioning and the cable is intact.
System control rotary switch cable	If the system control rotary switch appears unresponsive (ALOM cannot read rotary switch position), but the Power button works and the system stays powered on, you should suspect either that this cable is loose or defective, or (less likely) that there is a problem with the system configuration card reader.

Note – Most replacement cables for the Netra 440 server are available only as part of a cable kit, Sun part number F595-7286.

Monitoring the System

Sun provides the Sun Advanced Lights Out Manager (ALOM) tool that can give you advance warning of difficulties and prevent future downtime.

This monitoring tool lets you specify system criteria that bear watching. For instance, you can enable alerts for system events (such as excessive temperatures, power supply or fan failures, system resets), and be notified if those events occur. Warnings can be reported by icons in the software's graphical user interface, or you can be notified by email whenever a problem occurs.

Monitoring the System Using Advanced Lights Out Manager

Advanced Lights Out Manager (ALOM) enables you to monitor and control your server over a serial port or a network interface. The ALOM system controller provides a command-line interface that enables you to administer the server from remote locations. This may be especially useful when servers are geographically distributed or physically inaccessible.

ALOM also lets you remotely access the system console and run diagnostics (like POST) that would otherwise require physical proximity to the server's serial port. ALOM can send email notification of hardware failures or other server events.

The ALOM system controller runs independently, and uses standby power from the server. Therefore, ALOM firmware and software continue to be effective when the server operating system goes offline, or when power to the server itself is turned off.

TABLE 2-6 lists the items that ALOM enables you to monitor on the Netra 440 server.

TABLE 2-6 What ALOM Monitors

Item Monitored	What ALOM Reveals	Command to Type
Hard drives	Whether each slot has a drive present, and whether the drive reports OK status	`showenvironment`
Fan trays	Fan speed and whether the fan trays report OK status	`showenvironment`
CPU/memory modules	The presence of a CPU/memory module and the temperature measured at each CPU, as well as any thermal warning	`showenvironment`
Operating system status	Whether the operating system is running, stopped, initializing, or in some other state	`showplatform`
Power supplies	Whether each bay has a power supply present, and whether the power supply reports OK status	`showenvironment`
System temperature	Ambient and CPU core temperatures as measured at several locations in the system, as well as any thermal warning	`showenvironment`
Server front panel	System control rotary switch position and status of LEDs	`showenvironment`
User sessions	Which users are logged in to ALOM, and through which connections	`showusers`

For instructions on using ALOM to monitor a Netra 440 system, see "Monitoring the System Using Sun Advanced Lights Out Manager" on page 68.

Exercising the System

It is relatively easy to detect when a system component fails outright. However, when a system has an intermittent problem or seems to be "behaving strangely," a software tool that stresses or exercises the computer's many subsystems can help disclose the source of the emerging problem and prevent long periods of reduced functionality or system downtime.

Sun provides two tools for exercising Netra 440 servers:

- SunVTS software
- Hardware Diagnostic Suite software

TABLE 2-7 shows the FRUs that each system exercising tool is capable of isolating. Note that individual tools do not necessarily test *all* the components or paths of a particular FRU.

TABLE 2-7 FRU Coverage of System–Exercising Tools

FRU	**SunVTS**	**Hardware Diagnostic Suite**
ALOM system controller card	✓	
Connector board assembly	No coverage. See TABLE 2-5 for fault isolation hints.	
CPU/memory module	✓	✓
DIMMs	✓	✓
Hard drive	✓	✓
DVD drive	✓	
Fan tray 3	No coverage. See TABLE 2-8 for fault isolation hints.	
Fan trays 0-2	No coverage. See TABLE 2-8 for fault isolation hints.	
Motherboard	✓	✓
Power supply	✓	
SCSI backplane	✓	
System configuration card reader	No coverage. See TABLE 2-5 for fault isolation hints.	
System configuration card	✓	

Some FRUs are not isolated by any system exercising tool.

TABLE 2-8 FRUs Not Directly Isolated by System–Exercising Tools

FRU	Diagnostic Hints
Connector board assembly	See TABLE 2-5.
DVD drive cable	See TABLE 2-5.
Fan tray 3	If this FRU fails, ALOM issues an alert message: `SC Alert: PCI_FAN @ FT0 Failed.`
Fan trays 0-2	If this FRU fails, ALOM issues an alert message: `SC Alert: CPU_FAN @ FT1 Failed.`
SCSI data cable	See TABLE 2-5.
Connector board power cable	See TABLE 2-5.

Exercising the System Using SunVTS Software

SunVTS software validation test suite performs system and subsystem stress testing. You can view and control a SunVTS session over a network. Using a remote machine, you can view the progress of a testing session, change testing options, and control all testing features of another machine on the network.

You can run SunVTS software in five different test modes:

- *Connection mode* – SunVTS software verifies the presence of device controllers on all subsystems. This typically takes no more than a few minutes and is a good "sanity check" of the system connections.
- *Functional mode* – SunVTS software exercises only the specific subsystems you choose. This is the default mode. In Functional mode, selected tests are run in parallel. This mode uses system resources heavily, so you should not run any other applications at the same time.
- *Auto Config mode* – SunVTS software automatically detects all subsystems and exercises them in one of two ways:
  - *Confidence testing* – SunVTS software performs one pass of tests on all subsystems, and then stops. For typical system configurations, this requires one or two hours.
  - *Comprehensive testing* – SunVTS software exhaustively and repeatedly tests all subsystems for up to 24 hours.
- *Exclusive mode* – SunVTS software exercises only the specific subsystems you choose. Selected tests are run one at a time. A few tests are *only* available in this mode, including: `l1dcachetest`, `l2cachetest`, `l2sramtest`, `mpconstest`, `mptest`, `systest`, `env6test`, `i2c2test`, and `ssptest`.

- *Online mode* – SunVTS software exercises only the specific subsystems you choose. Selected tests are run one at a time until one complete system pass is achieved. This mode is useful for performing tests while other applications are running.

Since SunVTS software can run many tests in parallel and can consume many system resources, you should take care when using it on a production system. If you are stress-testing a system using SunVTS software's Comprehensive test mode, you should not run anything else on that system at the same time.

The Netra 440 server to be tested must be up and running if you want to use SunVTS software, since it relies on the Solaris OS. Since SunVTS software packages are optional, they may not be installed on your system. Turn to "Checking Whether SunVTS Software Is Installed" on page 90 for instructions.

It is important to use the most up-to-date version of SunVTS available, to ensure that you have the latest suite of tests. You can download the most recent SunVTS software from `http://www.sun.com/oem/products/vts/`.

For instructions on running SunVTS software to exercise the Netra 440 server, see "Exercising the System Using SunVTS Software" on page 86. For more information about the product, refer to:

- *SunVTS User's Guide* – Describes SunVTS features as well as how to start and control the various user interfaces.
- *SunVTS Test Reference Manual* – Describes each SunVTS test, option, and command-line argument.
- *SunVTS Quick Reference Card* – Gives an overview of the main features of the graphical user interface (GUI).
- *SunVTS Documentation Supplement* – Describes the latest product enhancements and documentation updates not included in the *SunVTS User's Guide* and *SunVTS Test Reference Manual*.

These documents are available on the Solaris Supplement CD and on the Web at: `http://www.sun.com/documentation`. You should also consult the SunVTS README file located at `/opt/SUNWvts/`. This document provides late-breaking information about the installed version of the product.

SunVTS Software and Security

During SunVTS software installation, you must choose between Basic or Sun Enterprise Authentication Mechanism (SEAM) security. Basic security uses a local security file in the SunVTS installation directory to limit the users, groups, and hosts permitted to use SunVTS software. SEAM security is based on Kerberos—the standard network authentication protocol—and provides secure user authentication, data integrity, and privacy for transactions over networks.

If your site uses SEAM security, you must have the SEAM client and server software installed in your networked environment and configured properly in both Solaris and SunVTS software. If your site does not use SEAM security, do not choose the SEAM option during SunVTS software installation.

If you enable the wrong security scheme during installation, or if you improperly configure the security scheme you chose, you may find yourself unable to run SunVTS tests. For more information, refer to the *SunVTS User's Guide* and the instructions accompanying the SEAM software.

Identifying Memory Modules

System firmware, including POST, has multiple ways of referring to memory. In most cases, such as when running tests or displaying configuration information, firmware refers to memory "banks." These are *logical* and not physical banks (see CODE EXAMPLE 2-19).

CODE EXAMPLE 2-19 POST Reference to Logical Memory Banks

```
0>Memory interleave set to 0
0>     Bank 0   512MB : 00000000.00000000 -> 00000000.20000000.
0>     Bank 1   512MB : 00000001.00000000 -> 00000001.20000000.
0>     Bank 2   512MB : 00000002.00000000 -> 00000002.20000000.
0>     Bank 3   512MB : 00000003.00000000 -> 00000003.20000000.
```

However, in POST error output (see CODE EXAMPLE 2-20), the firmware provides a memory slot identifier (`B0/D1 J0602`**)**. Note that `B0/D1` identifies the memory slot and is visible on the circuit board when the DIMM is installed. The label `J0602` also identifies the memory slot, but is not visible unless you remove the DIMM from the slot.

CODE EXAMPLE 2-20 POST Reference to Physical ID and Logical Bank

```
1>H/W under test = CPU3 B0/D1 J0602 side 1 (Bank 1), CPU Module C3
```

Adding to the potential confusion, when configuring system memory, you must also contend with the separate notion of *physical memory banks*: DIMMs must be installed as pairs of the same capacity and type within each physical bank.

The following sections clarify how memory is identified.

Physical Identifiers

Each CPU/memory module's circuit board contains silk-screened labels that uniquely identify every DIMM on that board. Each label is in this form:

B*x*/D*y*

Where *x* indicates the physical bank, and *y* the DIMM number within the bank.

In addition, a "J" number silk-screened on the circuit board uniquely identifies each DIMM slot. However, this slot number is not readily visible unless the DIMM is removed from the slot.

If you run POST and it finds a memory error, the error message will include the physical ID of the failed DIMM and the "J" number of the failed DIMM's slot, making it easy to determine which parts you need to replace.

Note – To ensure compatibility and maximize system uptime, you should replace DIMMs in pairs. Treat both DIMMs in a physical bank as one FRU.

Logical Banks

Logical banks reflect the system's internal memory architecture and not the architecture of the system's field-replaceable units. In the Netra 440 server, each logical bank spans two physical DIMMs. Since firmware-generated status messages refer only to logical banks, it is not possible to use these status messages to isolate a memory problem to a single failed DIMM. POST error messages, on the other hand, specify failures to the FRU level.

Note – To isolate faults in the memory subsystem, run POST diagnostics.

Correspondence Between Logical and Physical Banks

TABLE 2-9 shows the logical-to-physical memory bank mapping for the Netra 440 server.

TABLE 2-9 Logical and Physical Memory Banks in a Netra 440 Server

Logical Bank (As Given in Firmware Output)	Physical Identifiers (As Shown on Circuit Board)	Physical Bank
Bank 0	B0/D0 and B0/D1	Bank 0
Bank 1		
Bank 2	B1/D0 and B1/D1	Bank 1
Bank 3		

FIGURE 2-4 depicts the same mapping graphically.

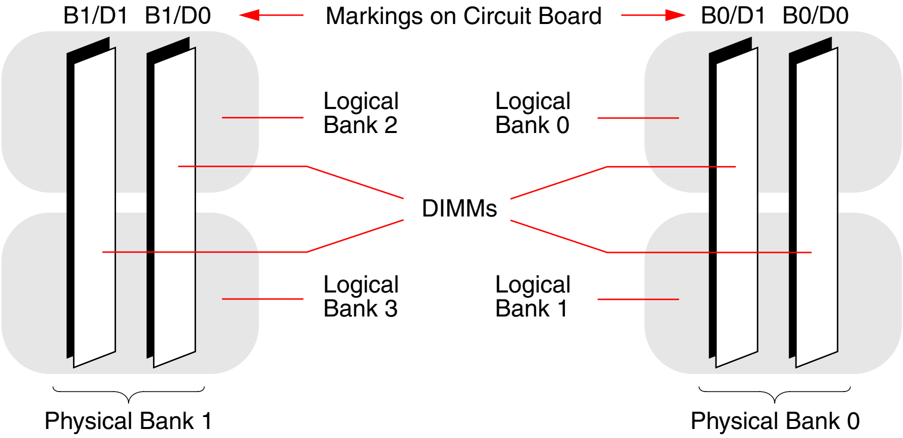


FIGURE 2-4 How Logical Memory Banks Map to DIMMs

Identifying CPU/Memory Modules

Since each CPU/memory module has its own set of DIMMs, you need to determine the CPU/memory module in which a faulty DIMM resides. This information is given in the POST error message:

```
1>H/W under test = CPU3 B0/D1 J0602 side 1 (Bank 1), CPU Module C3
```

In this example, the cited module is `CPU Module C3`.

The processors are numbered according to the slot in which they are installed, and these slots are numbered 0 to 3, left to right, as you look down on the Netra 440 server's chassis from the front (see FIGURE 2-5).

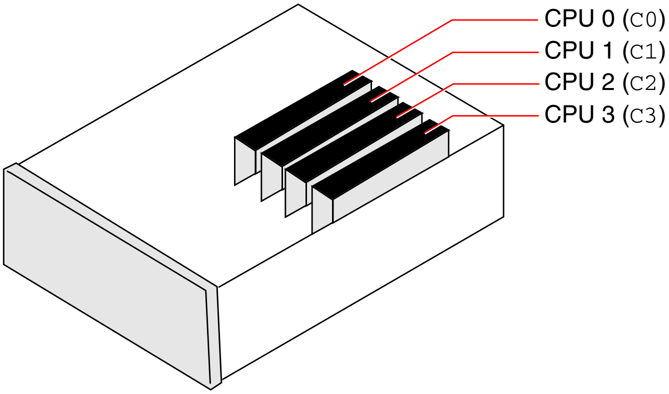


FIGURE 2-5 CPU/Memory Module Numbering

For example, if a Netra 440 server has only two CPU/memory modules installed, and if those are located in the leftmost and rightmost slots, then the firmware will refer to the two system processors as CPU 0 and CPU 3.

The failed DIMM called out by the previous POST error message, then, resides in the rightmost CPU/memory module (C3), and is labeled B0/D1 on that module's circuit board.

OpenBoot Diagnostics Test Descriptions

This section describes the OpenBoot Diagnostics tests and commands available to you. For background information about these tests, see "OpenBoot Diagnostics Tests" on page 15.

TABLE 2-10 OpenBoot Diagnostics Menu Tests

Test Name	What It Does	FRU(s) Tested
`flashprom@2,0`	Performs a checksum test on the boot PROM.	Motherboard
`i2c@0,320`	Tests the I^2C environmental monitoring subsystem, which includes various temperature and other sensors located on the motherboard and on other FRUs.	Motherboard, power supplies, SCSI disks, CPU/memory modules
`ide@d`	Tests the on-board IDE controller and IDE bus subsystem that controls the DVD-ROM drive.	Motherboard, DVD-ROM drive
`network@1`	Tests the on-board Ethernet controller, running internal loopback tests. Can also run external loopback tests, but only if you install a loopback connector (not provided).	Motherboard
`network@2`	Same as above, for the other on-board Ethernet controller.	Motherboard
`rmc-comm@0,3e8`	Tests communication with the ALOM system controller, and requests that ALOM diagnostics run.	ALOM card
`rtc@0,70`	Tests the registers of the real-time clock and verifies that it is running.	Motherboard
`scsi@2`	Tests internal SCSI hard drives.	Motherboard, SCSI backplane, SCSI disks
`scsi@2,1`	Tests any external SCSI hard drives attached.	Motherboard, SCSI cable, SCSI disks

`serial@0,3f8`
`serial@0,2e8`
	Tests all possible baud rates supported by the `ttya` and `ttyb` serial lines. Performs internal and external loopback tests on each line at each speed.	Motherboard

`usb@a` `usb@b`	Tests the writable registers of the USB open host controller.	Motherboard

TABLE 2-11 describes the commands you can type from the `obdiag>` prompt.

TABLE 2-11 OpenBoot Diagnostics Test Menu Commands

Command	Description
`exit`	Exits OpenBoot Diagnostics tests and returns to the `ok` prompt.
`help`	Displays a brief description of each OpenBoot Diagnostics command and OpenBoot configuration variable.
`set-default` *variable*	Restores the default value of an OpenBoot configuration variable.
`setenv` *variable value*	Sets the value for an OpenBoot configuration variable (also available from the `ok` prompt).
`test-all`	Tests all devices displayed in the OpenBoot Diagnostics test menu (also available from the `ok` prompt).
`test #`	Tests only the device identified by the menu entry number. (A similar function is available from the `ok` prompt. See "From the `ok` Prompt: The `test` and `test-all` Commands" on page 19.)
`test #,#`	Tests only the devices identified by the menu entry numbers.
`except #,#`	Tests all devices in the OpenBoot Diagnostics test menu except those identified by the menu entry numbers.
`what #,#`	Displays selected properties of the devices identified by the menu entry numbers. The information provided varies according to device type.

Decoding I^{2}C Diagnostic Test Messages

TABLE 2-12 describes each I^{2}C device in a Netra 440 server, and helps you associate each I^{2}C address with the proper FRU. For more information about I^{2}C tests, see "I^{2}C Bus Device Tests" on page 20.

TABLE 2-12 I^{2}C Bus Devices in a Netra 440 Server

Address	Associated FRU	What the Device Does
`alarm-fru-prom@0,ac`	Dry Contact Alarm	Dry Contact Alarm Board FRUID
`clock-generator@0,d2`	Motherboard	Controls PCI bus clock
`cpu-fru-prom@0,be`	CPU 0	Contains FRU configuration information

TABLE 2-12 I^{2}C Bus Devices in a Netra 440 Server *(Continued)*

Address	Associated FRU	What the Device Does
cpu-fru-prom@0,ce	CPU 1	Contains FRU configuration information
cpu-fru-prom@0,de	CPU 2	Contains FRU configuration information
cpu-fru-prom@0,ee	CPU 3	Contains FRU configuration information
dimm-spd@0,b6	CPU/memory module 0, DIMM 0	Contains FRU configuration information
dimm-spd@0,b8	CPU/memory module 0, DIMM 1	Contains FRU configuration information
dimm-spd@0,ba	CPU/memory module 0, DIMM 2	Contains FRU configuration information
dimm-spd@0,bc	CPU/memory module 0, DIMM 3	Contains FRU configuration information
dimm-spd@0,c6	CPU/memory module 1, DIMM 0	Contains FRU configuration information
dimm-spd@0,c8	CPU/memory module 1, DIMM 1	Contains FRU configuration information
dimm-spd@0,ca	CPU/memory module 1, DIMM 2	Contains FRU configuration information
dimm-spd@0,cc	CPU/memory module 1, DIMM 3	Contains FRU configuration information
dimm-spd@0,d6	CPU/memory module 2, DIMM 0	Contains FRU configuration information
dimm-spd@0,d8	CPU/memory module 2, DIMM 1	Contains FRU configuration information
dimm-spd@0,da	CPU/memory module 2, DIMM 2	Contains FRU configuration information
dimm-spd@0,dc	CPU/memory module 2, DIMM 3	Contains FRU configuration information
dimm-spd@0,e6	CPU/memory module 3, DIMM 0	Contains FRU configuration information
dimm-spd@0,e8	CPU/memory module 3, DIMM 1	Contains FRU configuration information
dimm-spd@0,ea	CPU/memory module 3, DIMM 2	Contains FRU configuration information
dimm-spd@0,ec	CPU/memory module 3, DIMM 3	Contains FRU configuration information
gpio@0,38	Power supply 0	PSU0 Status/Control REG

TABLE 2-12 I^{2}C Bus Devices in a Netra 440 Server *(Continued)*

Address	Associated FRU	What the Device Does
`gpio@0,3a`	Power supply 1	PSU1 Status/Control REG
`gpio@0,3c`	Power Distribution Board	PSU0_1 Status/Control REG
`gpio@0,42`	SCSI backplane	Indicates rotary switch status and drives Activity LEDs
`gpio@0,44`	Motherboard	Indicates power supply and CPU status
`gpio@0,46`	SCSI backplane	Indicates disk status and drives fault and Ok-to-Remove indicators
`gpio@0,48`	Motherboard	Drives system LEDs and CPU overtemperature indication
`gpio@0,e0`	Power Supply 2	PSU2 Status/Control REG
`gpio@0,e2`	Power Supply 3	PSU3 Status/Control REG
`gpio@0,e4`	Power Distribution Board	PSU2_3 Status/Control REG
`hardware-monitor@0,5c`	Motherboard	Monitors temperatures, voltages, and fan speeds
`i2c-bridge@0,16`	Motherboard	Translates I^{2}C bus addresses and isolates bus devices
`i2c-bridge@0,18`	Motherboard	Translates I^{2}C bus addresses and isolates bus devices
`motherboard-fru-prom@0,a2`	Motherboard	Contains FRU configuration information
`pdb-fru-prom@0,7c`	Power Distribution Board	PDB FRUID
`power-supply-fru-prom@0,70`	Power Supply 2	PSU2 FRUID
`power-supply-fru-prom@0,72`	Power Supply 3	PSU3 FRUID
`power-supply-fru-prom@0,a4`	Power supply	Contains FRU configuration information
`power-supply-fru-prom@0,c0`	Power supply 0	PSU0 FRUID
`power-supply-fru-prom@0,c2`	Power supply 1	PSU1 FRUID
`rmc-fru-prom@0,a6`	ALOM card	Contains FRU configuration information
`scsi-fru-prom@0,a8`	SCSI backplane	Contains FRU configuration information
`temperature-sensor@0,9c`	SCSI backplane	Senses system ambient temperature
`temperature@0,30`	CPU 0	Senses CPU die temperature
`temperature@0,64`	CPU 1	Senses CPU die temperature
`temperature@0,80`	CPU 2	Senses CPU die temperature
`temperature@0,90`	CPU 3	Senses CPU die temperature

Terms in Diagnostic Output Terms

The status and error messages displayed by POST diagnostics and OpenBoot Diagnostics tests occasionally include acronyms or abbreviations for hardware subcomponents. TABLE 2-13 is included to assist you in decoding this terminology and associating the terms with specific FRUs, where appropriate.

TABLE 2-13 Abbreviations or Acronyms in Diagnostic Output

Term	Description	Associated FRU(s)
ADC	Analog-to-Digital Converter	Motherboard
APC	Advanced Power Control – A function provided by the Southbridge integrated circuit	Motherboard
Bell	A repeater circuit element that forms part of the system bus	Motherboard
CRC	Cyclic Redundancy Check	Not applicable
DMA	Direct Memory Access – In diagnostic output, usually refers to a controller on a PCI card	PCI card
HBA	Host Bus Adapter	Motherboard, various others
I^2C	Inter-Integrated Circuit (also written as I2C) – A bidirectional, two-wire serial data bus. Used mainly for environmental monitoring and control	Various, see TABLE 2-12
IO-Bridge	System bus to PCI bridge integrated circuit (same as "Tomatillo")	Motherboard
JBus	The system interconnect architecture—that is, the data and address buses	Motherboard
JTAG	Joint Test Access Group – An IEEE subcommittee standard (1149.1) for scanning system components	Not applicable
MAC	Media Access Controller – Hardware address of a device connected to a network	Motherboard
MII	Media Independent Interface – Part of the Ethernet controller	Motherboard
NVRAM	Refers to the system configuration card (SCC)	System configuration card
OBP	Refers to OpenBoot firmware	Not applicable
PHY	Physical Interface – Part of the Ethernet control circuit	Motherboard

TABLE 2-13 Abbreviations or Acronyms in Diagnostic Output *(Continued)*

Term	Description	Associated FRU(s)
POST	Power-On Self-Test	Not applicable
RTC	Real-Time Clock	Motherboard
RX	Receive – Communication protocol	Motherboard
Scan	A means for monitoring and altering the content of ASICs and system components, as provided for in the IEEE 1149.1 standard	Not applicable
Southbridge	Integrated circuit that controls the ALOM UART port and more	Motherboard
Tomatillo	System bus to PCI bridge integrated circuit	Motherboard
TX	Transmit – Communication protocol	Motherboard
UART	Universal Asynchronous Receiver Transmitter – Serial port hardware	Motherboard, ALOM card
UIE	Update-ended Interrupt Enable – A function provided by the real-time clock	Motherboard
XBus	A byte-wide bus for low-speed devices	Motherboard

CHAPTER 3

Isolating Failed Parts

The most important use of diagnostic tools is to isolate a failed hardware component so that you can quickly remove and replace it. Because servers are complex machines with many failure modes, no single diagnostic tool can isolate all hardware faults under all conditions. However, Sun provides a variety of tools that can help you discern what component needs replacing.

This chapter guides you in choosing the best tools and describes how to use these tools to reveal a failed part in your Netra 440 server. It also explains how to use the Locator LED to isolate a failed system in a large equipment room.

Topics covered in this chapter include:

- "Viewing and Setting OpenBoot Configuration Variables" on page 50
- "Operating the Locator LED" on page 51
- "Putting the System in Diagnostics Mode" on page 52
- "Bypassing Firmware Diagnostics" on page 54
- "Bypassing Diagnostics Temporarily" on page 55
- "Maximizing Diagnostic Testing" on page 56
- "Isolating Faults Using LEDs" on page 57
- "Isolating Faults Using POST Diagnostics" on page 60
- "Isolating Faults Using Interactive OpenBoot Diagnostics Tests" on page 62
- "Viewing Diagnostic Test Results After the Fact" on page 64
- "Choosing a Fault Isolation Tool" on page 65

If you want *background information* about the tools, turn to the section "Isolating Faults in the System" on page 32.

Note – Many of the procedures in this chapter assume that you are familiar with the OpenBoot firmware and that you know how to access the `ok` prompt. For background information, refer to the *Netra 440 Server System Administration Guide*.

Viewing and Setting OpenBoot Configuration Variables

Switches and OpenBoot configuration variables stored in the system configuration card determine how and when power-on self-test (POST) diagnostics and OpenBoot Diagnostics tests are performed. This section explains how to access and modify OpenBoot configuration variables. For a list of important OpenBoot configuration variables, see TABLE 2-1.

▼ To View and Set OpenBoot Configuration Variables

1. **Suspend the server's operating system software to reach the `ok` prompt.**

2. **Enter the following commands:**

   - To display the current values of all OpenBoot configuration variables, use the `printenv` command.

   The following example shows a short excerpt of this command's output.

```
ok printenv
Variable Name         Value                         Default Value

diag-level            min                            min
diag-switch?          false                          false
```

   - To set or change the value of an OpenBoot configuration variable, use the `setenv` command.

```
ok setenv diag-level max
diag-level =         max
```

   - To set OpenBoot configuration variables that accept multiple keywords, separate keywords with a space.

```
ok setenv post-trigger power-on-reset error-reset
post-trigger = power-on-reset error-reset
```

Note – The `test-args` variable operates differently from other OpenBoot configuration variables. It requires a single argument consisting of a comma-separated list of keywords. For details, see "Controlling OpenBoot Diagnostics Tests" on page 16.

Changes to OpenBoot configuration variables usually take effect on the next reboot.

Operating the Locator LED

The Locator LED helps you to quickly find a specific system among numerous systems in a room. For background information about system LEDs, see the *Netra 440 Server System Administration Guide* (817-3884-xx).

You can turn the Locator LED on and off either from the system console, or by using the Advanced Lights Out Manager (ALOM) command-line interface.

▼ To Operate the Locator LED

1. **Access either the system console or the system controller.**

   For instructions, refer to the *Netra 440 Server System Administration Guide*.

2. **Determine the current state of the Locator LED.**

   Do one of the following:

   - **From the system console, type:**

```
# /usr/sbin/locator
The 'system' locator is on
```

   - **From the ALOM system controller, type:**

```
sc> showlocator
Locator LED is ON
```

3. **Turn the Locator LED *on*.**

   Do one of the following:

   - **From the system console, type:**

```
# /usr/sbin/locator -n
```

   - **From the ALOM system controller, type:**

```
sc> setlocator on
```

4. **Turn the Locator LED *off*.**

   Do one of the following:

   - **From the system console, type:**

```
# /usr/sbin/locator -f
```

   - **From the system controller, type:**

```
sc> setlocator off
```

Putting the System in Diagnostics Mode

Firmware-based diagnostic tests can be bypassed to expedite the server's startup process. The following procedure ensures that POST and OpenBoot Diagnostics tests *do run* during startup. For background information, see "Diagnostics: Reliability versus Availability" on page 14.

To Put the System In Diagnostics Mode

1. **Log in to the system console and access the** `ok` **prompt.**

2. **Do one of the following, whichever is more convenient:**

   - **Set the server's system control rotary switch to the Diagnostics position.**

     You can do this at the machine's front panel or, if you are running your test session remotely from console display, through the ALOM interface.

   - **Set the** `diag-switch?` **variable. Type:**

```
ok setenv diag-switch? true
```

3. **Set the OpenBoot configuration** `diag-script` **variable to** `normal`**. Type:**

```
ok setenv diag-script normal
```

This allows OpenBoot Diagnostics tests to run automatically on all motherboard components.

Note – If you prefer that OpenBoot Diagnostics examine all IEEE 1275-compatible devices (not just those on the motherboard), set the `diag-script` variable to `all`.

---

4. **Set OpenBoot configuration variables to trigger diagnostic tests. Type:**

```
ok setenv post-trigger power-on-reset error-reset
ok setenv obdiag-trigger power-on-reset error-reset
```

5. **Set the maximum POST diagnostic test level. Type:**

```
ok setenv diag-level max
```

This ensures the most thorough power-on self-test possible. The maximum testing level requires considerably longer to complete than the minimum. Depending on system configuration, you may need to wait an additional 10 to 20 minutes for the server to boot.

Bypassing Firmware Diagnostics

POST and OpenBoot Diagnostics tests can be bypassed to expedite the server's startup process. For background information, see "Diagnostics: Reliability versus Availability" on page 14.

Caution – Bypassing diagnostic tests sacrifices system reliability by allowing a system to attempt to boot when it may have a serious hardware problem.

▼ To Bypass Firmware Diagnostics

1. **Log in to the system console and access the `ok` prompt.**

2. **Ensure that the server's system control rotary switch is set to the Normal position.**

   Setting the rotary switch to the Diagnostics position overrides the OpenBoot configuration variable settings and causes diagnostic tests to run.

3. **Turn off the `diag-switch?` and `diag-script` variables. Type:**

```
ok setenv diag-switch? false
ok setenv diag-script none
```

4. **Set OpenBoot configuration trigger variables to bypass diagnostics. Type:**

```
ok setenv post-trigger none
ok setenv obdiag-trigger none
```

The Netra 440 server is now configured to minimize the time it takes to reboot. If you change your mind and want to force diagnostic tests to run, see "Putting the System in Diagnostics Mode" on page 52.

Bypassing Diagnostics Temporarily

The ALOM system controller provides a “back-door” method of skipping diagnostic tests and booting the system. This procedure is only of assistance in those unusual circumstances where:

- The system is configured to run diagnostic tests automatically on power up.
- The hardware is functional and capable of booting, but is precluded from doing so by a firmware malfunction or incompatibility.

To Bypass Diagnostics Temporarily

1. **Log in to the ALOM system controller and access the `sc>` prompt.**

2. **Type the following command:**

```
sc> bootmode skip_diag
```

This command temporarily configures the system to skip its firmware-based diagnostic tests, regardless of how the OpenBoot configuration variables are set.

3. **Within 10 minutes, power cycle the system. Type:**

```
sc> poweroff
Are you sure you want to power off the system [y/n]? y
sc> poweron
```

You must execute the above commands within 10 minutes of using ALOM to change the boot mode. Ten minutes after you issue the ALOM `bootmode` command, the system reverts back to its default boot mode as governed by the current settings of OpenBoot configuration variables, including `diag-switch`, `post-trigger`, and `obdiag-trigger`.

For more information about OpenBoot configuration variables and how they affect diagnostics, see “Controlling POST Diagnostics” on page 13.

If you suspect an incompatible or corrupted firmware image caused the problems you observed with firmware diagnostics, you should now restore the system firmware to a reliable state.

For more information about restoring the system firmware, contact your authorized service provider.

Maximizing Diagnostic Testing

To maximize system reliability, it is useful to have POST and OpenBoot Diagnostics tests trigger in the event of an operating system panic or any reset, and to run automatically the most comprehensive tests possible. For background information, see "Diagnostics: Reliability versus Availability" on page 14.

▼ To Maximize Diagnostic Testing

1. **Log in to the system console and access the** `ok` **prompt.**

2. **Do one of the following, whichever is more convenient:**

   - **Set the server's system control rotary switch to the Diagnostics position.**

     You can do this at the server's front panel or, if you are running your test session remotely from console display, through the ALOM interface.

   - **Set the** `diag-switch?` **variable to** `true`**. Type:**

```
ok setenv diag-switch? true
```

3. **Set the OpenBoot configuration** `diag-script` **variable to** `all`**. Type:**

```
ok setenv diag-script all
```

This allows OpenBoot Diagnostics tests to run automatically on all motherboard components and IEEE 1275-compatible devices.

Note – If you prefer that OpenBoot Diagnostics examine only motherboard-based devices, set the `diag-script` variable to `normal`.

---

4. **Set OpenBoot configuration variables to trigger diagnostic tests. Type:**

```
ok setenv post-trigger all-resets
ok setenv obdiag-trigger all-resets
```

5. **Set the maximum POST diagnostic test level. Type:**

```
ok setenv diag-level max
```

This ensures the most thorough testing possible. The maximum testing level requires considerably longer to complete than the minimum. Depending on system configuration, you may need to wait an additional 10 to 20 minutes for the server to boot.

Isolating Faults Using LEDs

While not a comprehensive diagnostic tool, LEDs located on the chassis and on selected system components can serve as front-line indicators of a limited set of hardware failures.

You can view LED status by direct inspection of the system's front and back panels. You can also view the status of certain LEDs from the ALOM system controller command-line interface.

Note – Most LEDs available on the front panel are also duplicated on the back panel.

▼ To Isolate Faults Using LEDs

1. **Check the system LEDs.**

   There is a group of three LEDs located near the top left corner of the front panel and duplicated on the back panel. Their status can tell you the following:

LED Name (location; color)	Indicates	Action
Locator (left; white)	A system administrator can turn this on to flag a system that needs attention.	Identify a particular system among many.
Service Required (middle; amber)	If lit, hardware or software has detected a problem with the system.	Check other LEDs or run diagnostics to determine the problem source.
System Activity (right; green)	If blinking, operating system is in the process of booting. If off, operating system has stopped.	Not applicable.

   The Locator and Service Required LEDs are powered by the system's 5-volt standby power source and remain lit for any fault condition that results in a system shutdown.

   Note – To view the status of system LEDs from ALOM, type `showenvironment` from the `sc>` prompt.

2. **Check the power supply LEDs.**

   Each power supply has a set of four LEDs located on the front panel and duplicated on the back panel. Their status can tell you the following:

LED Name (location; color)	Indicates	Action
OK-to-Remove (top; blue)	If lit, power supply can safely be removed.	Remove power supply as needed.
	Note: Remove a failed power supply only when you are ready to install its replacement. Both power supplies must remain in place to ensure proper air circulation and chassis cooling.	
Service Required (2nd from top; amber)	If lit, there is a problem with the power supply or its internal fan.	Replace the power supply.
Power OK (3rd from top; green)	If off, inadequate DC power is being produced by the supply.	Remove and reseat the power supply. If this does not help, replace the supply.
Standby Available (bottom; green)	If off, either AC power is not reaching the supply, or the supply is not producing adequate 5V standby power.	Check the power cord and the outlet to which it connects. If necessary, replace the supply.

3. **Check the hard drive LEDs.**

   Hard drive LEDs are located behind the left system door. Just to the right of each hard drive is a set of three LEDs. Their status can tell you the following:

LED Name (location; color)	Indicates	Action
OK-to-Remove (top; blue)	If lit, disk can safely be removed.	Remove disk as needed.
Service Required (middle; amber)	This LED is reserved for future use.	Not applicable.
Activity (bottom; green)	If lit or blinking, disk is operating normally.	Not applicable.

4. **Check the DVD-ROM LED.**

   The DVD-ROM drive features a Power/Activity LED that tells you the following:

LED Name (color)	Indicates	Action
Power/Activity (green)	If lit or blinking, drive is operating normally.	If this LED is off, and you know the system is receiving power, check the DVD-ROM drive and its cables.

5. **Check the Ethernet port LEDs.**

   Two Ethernet port LEDs are located on the system back panel.

LED Name (color)	Indicates	Action
Link/Activity (green)	If lit, a link is established. If blinking, there is activity. Both states indicate normal operation.	If this LED is off and you know a link is being attempted, check the Ethernet cables.
Speed (amber)	If lit, a Gigabit Ethernet connection is established. If off, a 10/100-Mbps Ethernet connection is established.	

6. **If LEDs do not disclose the source of a suspected problem, try putting the affected server in Diagnostics mode.**

   See "Putting the System in Diagnostics Mode" on page 52.

   You can also run power-on self-test (POST) diagnostics. See "Isolating Faults Using POST Diagnostics" on page 60.

Isolating Faults Using POST Diagnostics

This section explains how to run power-on self-test (POST) diagnostics to isolate faults in a Netra 440 server. For background information about POST diagnostics and the boot process, see Chapter 2.

To Isolate Faults Using POST Diagnostics

1. **Log in to the system console and access the `ok` prompt.**

   This procedure assumes that the system is in diagnostics mode. See:

   - "Putting the System in Diagnostics Mode" on page 52

   The procedure also assumes that the system console is in its default configuration, so that you are able to switch between the system controller and the system console. Refer to the *Netra 440 Server System Administration Guide*.

2. **(Optional) Set the OpenBoot configuration variable `diag-level` to `max`. Type:**

```
ok setenv diag-level max
diag-level =     max
```

   This provides the most extensive diagnostic testing.

3. **Power on the server.**

   Do one of the following:

   - **Press the Power button at the server's front panel.**
   - **Access the ALOM system controller and type:**

```
ok #.
sc>
```

   Then, from the `sc>` prompt, type:

```
sc> poweron
sc> console
ok
```

   The system runs the POST diagnostics and displays status and error messages through the local serial terminal.

   Note – You will not see any POST output if you remain at the `sc>` prompt. You must return to the `ok` prompt by typing the `console` command as shown above.

4. **Examine the POST output.**

   Each POST error message includes a "best guess" as to which field-replaceable unit (FRU) was the source of failure. In some cases, there may be more than one possible source, and these are listed in order of decreasing likelihood.

Note – Should the POST output contain code names and acronyms with which you are unfamiliar, see TABLE 2-13 in "Terms in Diagnostic Output Terms" on page 47.

5. **Try replacing the FRU or FRUs indicated by POST error messages, if any.**

   For replacement instructions, refer to the *Netra 440 Server Service Manual.*

6. **If the POST diagnostics did not turn up any problems, but your system does not start up, try running the interactive OpenBoot Diagnostics tests.**

   For more information, see:

   - "Isolating Faults Using Interactive OpenBoot Diagnostics Tests" on page 62

Isolating Faults Using Interactive OpenBoot Diagnostics Tests

Because OpenBoot Diagnostics tests require access to some of the same hardware resources used by the operating system, the tests cannot be run reliably after an operating system halt or L1-A (Stop-A) key sequence. You need to reset the system before running OpenBoot Diagnostics tests, and then reset the system again after testing. Instructions for doing this follow.

▼ To Isolate Faults Using Interactive OpenBoot Diagnostics Tests

1. **Log in to the system console and access the `ok` prompt.**

2. **Set the `auto-boot?` OpenBoot configuration variable to `false`. Type:**

```
ok setenv auto-boot? false
```

3. **Reset or power cycle the system.**

4. **Invoke the OpenBoot Diagnostics tests. Type:**

```
ok obdiag
```

The `obdiag>` prompt and test menu appear. The menu is shown in FIGURE 2-3.

5. ***(Optional)* Set the desired test level.**

   You may want to perform the most extensive testing possible by setting the `diag-level` OpenBoot configuration variable to `max`:

```
obdiag> setenv diag-level max
```

Note – If `diag-level` is set to `off`, OpenBoot firmware returns a passed status for all core tests, but performs no testing.

   You can set any OpenBoot configuration variable (see TABLE 2-1) from the `obdiag>` prompt in the same way.

6. **Type the appropriate command and numbers for the tests you want to run.**

   For example, to run all available OpenBoot Diagnostics tests, type:

```
obdiag> test-all
```

   To run a particular test, type:

```
obdiag> test  #
```

   Where # represents the number of the desired test.

   For a list of OpenBoot Diagnostics test commands, see "Interactive OpenBoot Diagnostics Commands" on page 18. The menu of numbered tests is shown in FIGURE 2-3.

7. **When you are done running OpenBoot Diagnostics tests, exit the test menu. Type:**

```
obdiag> exit
```

   The `ok` prompt reappears.

8. **Set the `auto-boot?` OpenBoot configuration variable back to `true`. Type:**

```
ok setenv auto-boot? true
```

   This allows the operating system to resume starting up automatically after future system resets or power cycles.

9. **To reboot the system, type:**

```
ok reset-all
```

The system stores the OpenBoot configuration variable settings and boots automatically when the `auto-boot?` variable is set to `true`.

Try replacing the FRU or FRUs indicated by OpenBoot Diagnostics error messages, if any. For FRU replacement instructions, refer to the *Netra 440 Server Service Manual.*

Viewing Diagnostic Test Results After the Fact

Summaries of the results from the most recent power-on self-test (POST) and OpenBoot Diagnostics tests are saved across power cycles.

▼ To View Diagnostic Test Results

1. **Log in to the system console and access the `ok` prompt.**
2. **View the desired category of test results.**
   - To see a summary of the last execution of POST diagnostics, type:

```
ok show-post-results
```

   - To see a summary of the last execution of OpenBoot Diagnostics tests, type:

```
ok show-obdiag-results
```

You should see a system-dependent list of hardware components, along with an indication of which components passed and which failed POST or OpenBoot Diagnostics tests.

Choosing a Fault Isolation Tool

This section helps you choose the right tool to isolate a failed part in a Netra 440 server. Consider the following questions when selecting a tool.

1. Have you checked the LEDs?

   Certain system components have built-in LEDs that can alert you when that component requires replacement. For detailed instructions, see "Isolating Faults Using LEDs" on page 57.

2. Does the system boot?

   - If the system *cannot* boot, you must run firmware-based diagnostics that do not depend on the operating system.
   - If the system *can* boot, you should use a more comprehensive tool. The typical fault isolation process is illustrated in FIGURE 3-1.

3. Do you intend to run the tests remotely?

   The ALOM system controller software enable you to run tests from a remote server. In addition, ALOM provides a means of redirecting system console output, allowing you to remotely view and run tests—like POST diagnostics—that usually require physical proximity to the serial port on the server's back panel.

   SunVTS software, a system exercising tool, also enables you to run tests remotely using either the product's graphical interface, or tty-mode through remote login or Telnet session.

4. Will the tool test the suspected sources of the problem?

   Perhaps you already have some idea of what the problem is. If so, you want to use a diagnostic tool capable of testing the suspected problem sources.

   - TABLE 2-4 tells you which replaceable hardware parts can be isolated by each fault–isolating tool.
   - TABLE 2-7 tells you which replaceable hardware parts are covered by each system–exercising tool.

5. Is the problem intermittent or software related?

   If a problem is not caused by a clearly defective hardware component, then you may want to use a system–exerciser tool rather than a fault–isolation tool. See Chapter 2 for instructions and "Exercising the System" on page 36 for background information.

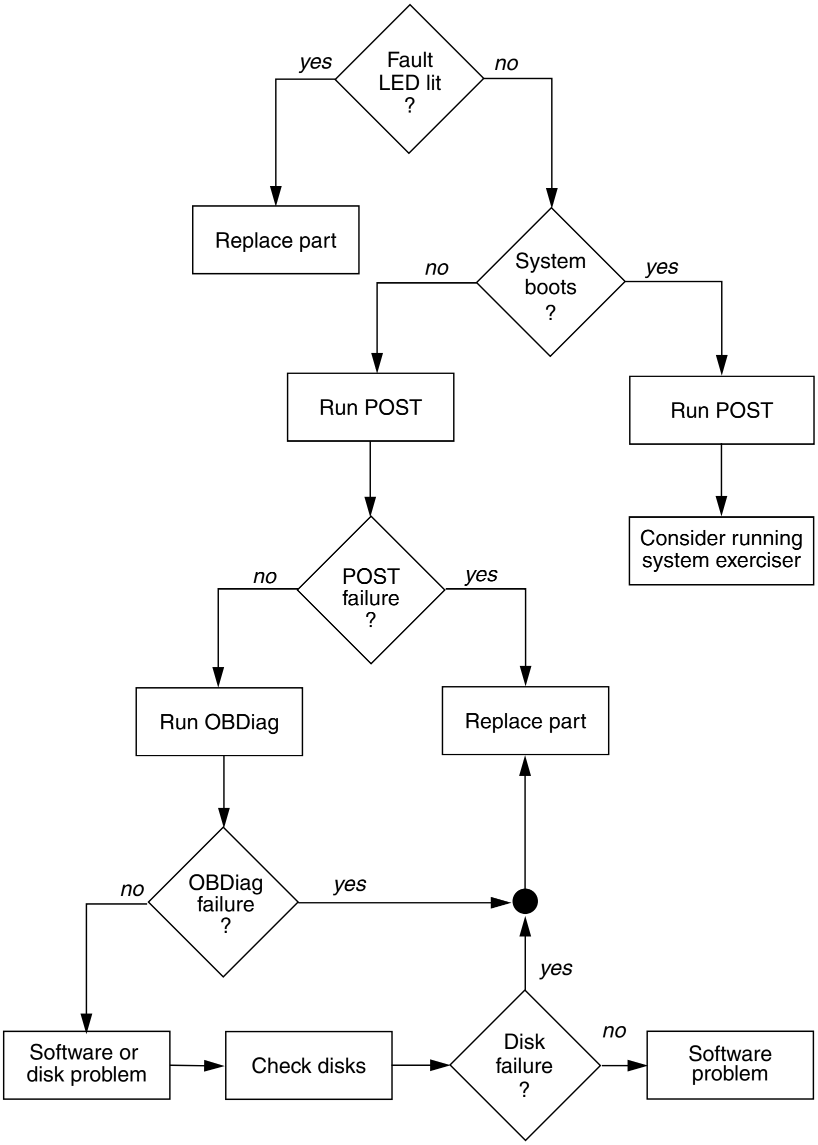


FIGURE 3-1 Choosing a Tool to Isolate Hardware Faults

CHAPTER 4

Monitoring the System

When something goes wrong with the system, diagnostic tools can help you figure out what caused the problem. Indeed, this is the principal use of most diagnostic tools. However, this approach is inherently reactive. It means waiting until a component fails outright.

Some diagnostic tools allow you to be more proactive by monitoring the system while it is still "healthy." Monitoring tools give administrators early warning of imminent failure, thereby allowing planned maintenance and better system availability. Remote monitoring also allows administrators the convenience of checking on the status of many machines from one centralized location.

Sun provides the Advanced Lights Out Manager (ALOM) software that you can use to monitor servers.

In addition to that tool, Sun provides software-based and firmware-based commands that display various kinds of system information. While not strictly monitoring tools, these commands enable you to review at a glance the status of different system aspects and components.

This chapter describes the tasks necessary to use these tools to monitor your Netra 440 server.

Tasks covered in this chapter include:

- "Monitoring the System Using Sun Advanced Lights Out Manager" on page 68
- "Using Solaris System Information Commands" on page 82
- "Using OpenBoot Information Commands" on page 83

If you want *background information* about the tools, turn to Chapter 2.

Note – Many of the procedures in this chapter assume that you are familiar with the OpenBoot firmware and that you know how to access the ok prompt. For background information, refer to the *Netra 440 Server System Administration Guide*.

Monitoring the System Using Sun Advanced Lights Out Manager

This section explains how to use Advanced Lights Out Manager (ALOM) to monitor a Netra 440 server, and steps you through some of the tool's most important features.

For background information about ALOM, see:

- "Monitoring the System Using Advanced Lights Out Manager" on page 35
- *Advanced Lights Out Manager Software User's Guide for the Netra 440 Server*.

There are several ways to connect to and use the ALOM system controller, depending on how your data center and its network are set up. This procedure assumes that you intend to monitor the Netra 440 system by way of an alphanumeric terminal or terminal server connected to the server's SERIAL MGT port, or by using a `telnet` connection to the NET MGT port.

The procedure also assumes that the system console is in its default configuration, so that you are able to switch between the system controller and the system console. Refer to the *Netra 440 Server System Administration Guide*.

▼ To Monitor the System Using Sun Advanced Lights Out Manager

1. **Log in to the system console and access the `ok` prompt.**

2. **If necessary, type the system controller escape sequence.**

   If you are not already seeing the `sc>` prompt, type the system controller escape sequence. By default, this sequence is `#.` (pound-period).

```
ok #.
```

3. **If necessary, log in to ALOM.**

   If you are not logged in to ALOM, you will be prompted to do so:

   ```
   Please login: admin
   Please Enter password: ******
   ```

   Enter the admin account login name and password, or the name and password of a different login account if one has been set up for you. For the purposes of this procedure, your account should have full privileges.

   Note – The first time you access ALOM, there is no admin account password. You are instructed to provide one the first time you attempt to execute a privileged command. Note the password you enter and retain it for future use.

   The sc> prompt appears:

   ```
   sc>
   ```

   This prompt indicates that you now have access to the ALOM system controller command-line interface.

4. **At the** sc> **prompt, type the** showenvironment **command.**

   ```
   sc> showenvironment
   ```

This command displays a great deal of useful data, starting with temperature readings from a number of thermal sensors.

CODE EXAMPLE 4-1 ALOM Reports on System Temperatures

```
=============== Environmental Status ===============


------------------------------------------------------------------------------
System Temperatures (Temperatures in Celsius):
------------------------------------------------------------------------------
Sensor        Status    Temp LowHard LowSoft LowWarn HighWarn HighSoft HighHard
------------------------------------------------------------------------------
C0.P0.T_CORE    OK         48    -20     -10       0      97      102      120
C1.P0.T_CORE    OK         53    -20     -10       0      97      102      120
C2.P0.T_CORE    OK         49    -20     -10       0      97      102      120
C3.P0.T_CORE    OK         57    -20     -10       0      97      102      120
C0.T_AMB        OK         28    -20     -10       0      70       82       87
C1.T_AMB        OK         33    -20     -10       0      70       82       87
C2.T_AMB        OK         27    -20     -10       0      70       82       87
C3.T_AMB        OK         28    -20     -10       0      70       82       87
MB.T_AMB        OK         32    -18     -10       0      65       75       85
```

Note – The warning and soft graceful shutdown thresholds noted in CODE EXAMPLE 4-1 are set at the factory and cannot be modified.

The sensors labeled `T_AMB` in CODE EXAMPLE 4-1 measure ambient temperatures at the CPU/memory modules, the motherboard, and the SCSI backplane. The sensors labeled `T_CORE` measure the internal temperatures of the processor chips themselves.

In the output shown in CODE EXAMPLE 4-1, `MB` refers to the motherboard, and `C`*n* refers to a particular CPU. For information about identifying CPU modules, see "Identifying CPU/Memory Modules" on page 41.

The showenvironment command also gives the position of the system control rotary switch and the condition of the three LEDs on the front panel.

CODE EXAMPLE 4-2 ALOM Reports on Rotary Switch Position and System Status LEDs

```
--------------------------------------
Front Status Panel:
--------------------------------------
Rotary Switch position: NORMAL

---------------------------------------------------
System Indicator Status:
---------------------------------------------------
SYS.LOCATE           SYS.SERVICE          SYS.ACT
--------------------------------------------------------
OFF                  OFF                  ON
```

The showenvironment command reports the status of system disks and fans.

CODE EXAMPLE 4-3 ALOM Reports on System Disks and Fans

```
--------------------------------------------
System Disks:
--------------------------------------------
Disk   Status            Service  OK2RM
--------------------------------------------
HDD0   OK                OFF      OFF
HDD1   OK                OFF      OFF
HDD2   OK                OFF      OFF
HDD3   OK                OFF      OFF

----------------------------------------------------------
Fans (Speeds Revolution Per Minute):
----------------------------------------------------------
Sensor           Status           Speed   Warn    Low
----------------------------------------------------------
FT0.F0.TACH      OK                3879   2400    750
FT1.F0.TACH      OK                3947   2400    750
FT2.F0.TACH      OK                4017   2400    750
FT3.F0           OK                  --     --     --
```

Voltage sensors located on the motherboard monitor important system voltages, and `showenvironment` reports these.

CODE EXAMPLE 4-4 ALOM Reports on Motherboard Voltages

```
------------------------------------------------------------------------------
Voltage sensors (in Volts):
------------------------------------------------------------------------------
Sensor         Status       Voltage LowSoft LowWarn HighWarn HighSoft
------------------------------------------------------------------------------
MB.V_+1V5      OK              1.49    1.20    1.27    1.72     1.80
MB.V_VCCTM     OK              2.53    2.00    2.12    2.87     3.00
MB.V_NET0_1V2D OK              1.26    0.96    1.02    1.38     1.44
MB.V_NET1_1V2D OK              1.26    0.96    1.02    1.38     1.44
MB.V_NET0_1V2A OK              1.26    0.96    1.02    1.38     1.44
MB.V_NET1_1V2A OK              1.25    0.96    1.02    1.38     1.44
MB.V_+3V3      OK              3.33    2.64    2.80    3.79     3.96
MB.V_+3V3STBY  OK              3.33    2.64    2.80    3.79     3.96
MB.BAT.V_BAT   OK              3.07      --    2.25      --       --
MB.V_SCSI_CORE OK              1.80    1.44    1.53    2.07     2.16
MB.V_+5V       OK              5.02    4.00    4.25    5.75     6.00
MB.V_+12V      OK             12.00    9.60   10.20   13.80    14.40
MB.V_-12V      OK            -11.96  -14.40  -13.80  -10.20    -9.60
```

Note – The warning and soft graceful shutdown thresholds noted in CODE EXAMPLE 4-4 are set at the factory and cannot be modified.

The showenvironment command tells you the status of each power supply, and the state of the LEDs located on each supply.

CODE EXAMPLE 4-5 ALOM Reports on Power Supply Status

```
--------------------------------------------
Power Supply Indicators:
--------------------------------------------
Supply    Active  Service  OK-to-Remove
--------------------------------------------
PS0       ON      OFF      OFF
PS1       ON      OFF      OFF
PS2       ON      OFF      OFF
PS3       ON      OFF      OFF

------------------------------------------------------------------------------
Power Supplies:
------------------------------------------------------------------------------
Supply  Status          Underspeed  Overtemp  Overvolt  Undervolt Overcurrent
------------------------------------------------------------------------------
PS0     OK              OFF         OFF       OFF       OFF        OFF
PS1     OK              OFF         OFF       OFF       OFF        OFF
PS2     OK              OFF         OFF       OFF       OFF        OFF
PS3     OK              OFF         OFF       OFF       OFF        OFF
```

This command reports on the status of motherboard circuit breakers (labeled MB.FF_SCSI*x*) and CPU module DC-to-DC converters (labeled C*n*.P0.FF_POK).

CODE EXAMPLE 4-6 ALOM Reports on Circuit Breakers and DC-to-DC Converters

```
----------------------
Current sensors:
----------------------
Sensor          Status
----------------------
MB.FF_SCSIA      OK
MB.FF_SCSIB      OK
MB.FF_POK        OK
C0.P0.FF_POK     OK
C1.P0.FF_POK     OK
C2.P0.FF_POK     OK
C3.P0.FF_POK     OK
```

Finally, this command tells you the status of the system alarms.

CODE EXAMPLE 4-7 ALOM Reports on System Alarms

```
--------------------------------------------
System Alarms:
--------------------------------------------
Alarm                   Relay          LED
--------------------------------------------
ALARM.CRITICAL          OFF            OFF
ALARM.MAJOR             OFF            OFF
ALARM.MINOR             OFF            OFF
ALARM.USER              OFF            OFF
```

5. **Type the** `showfru` **command.**

```
sc> showfru
```

This command, like the Solaris OS command `prtfru -c`, displays static FRU-ID information as available for several system FRUs. The specific information provided includes the date and location of manufacture, and the Sun part number.

CODE EXAMPLE 4-8 ALOM Reports on FRU Identification Information

```
FRU_PROM at PSO.SEEPROM
  Timestamp: MON SEP 16 16:47:05 2002
  Description: PWR SUPPLY, SYSTEM,75%-EFF,H-P
  Manufacture Location: DELTA ELECTRONICS CHUNGLI TAIWAN
  Sun Part No: 3001501
  Sun Serial No: T00065
  Vendor JDEC code: 3AD
  Initial HW Dash Level: 01
  Initial HW Rev Level: 02
  Shortname: PS
```

6. **Type the** `showlogs` **command.**

```
sc> showlogs
```

This command shows a history of noteworthy system events, the most recent being listed last.

CODE EXAMPLE 4-9 ALOM Reports on Logged Events

```
FEB 28 19:45:06 myhost: 0006001a: "SC Host Watchdog Reset Disabled"
FEB 28 19:45:06 myhost: 00060003: "SC System booted."
FEB 28 19:45:43 myhost: 00060000: "SC Login: User admin Logged on."
FEB 28 19:45:51 myhost: 0004000e: "SC Request to Power Off Host Immediately."
FEB 28 19:45:55 myhost: 00040002: "Host System has Reset"
FEB 28 19:45:56 myhost: 00040029: "Host system has shut down."
FEB 28 19:46:16 myhost: 00040001: "SC Request to Power On Host."
FEB 28 19:46:18 myhost: 0004000b: "Host System has read and cleared bootmode."
FEB 28 19:55:17 myhost: 00060000: "SC Login: User admin Logged on."
FEB 28 19:56:59 myhost: 00060000: "SC Login: User admin Logged on."
FEB 28 20:27:06 myhost: 0004004f: "Indicator SYS_FRONT.ACT is now ON"
FEB 28 20:40:47 myhost: 00040002: "Host System has Reset"
```

Note – The ALOM log messages are written into a so-called “circular buffer” of limited length (64 kilobytes). Once the buffer is filled, the oldest messages are overwritten by the newest ones.

7. **Examine the ALOM run log. Type:**

```
sc> consolehistory run -v
```

This command shows the log containing the most recent system console output from POST, OpenBoot PROM, and Solaris boot messages. In addition, this log records output from the server’s operating system.

CODE EXAMPLE 4-10 `consolehistory run -v` Command Output

```
May  9 14:48:22 Sun-SFV440-a rmclomv: SC Login: User admin Logged on.

#
# init 0
#
INIT: New run level: 0
The system is coming down.  Please wait.
System services are now being stopped.
Print services stopped.
```

CODE EXAMPLE 4-10 `consolehistory run -v` Command Output *(Continued)*

```
May  9 14:49:18 Sun-SFV440-a last message repeated 1 time

May  9 14:49:38 Sun-SFV440-a syslogd: going down on signal 15

The system is down.
syncing file systems... done
Program terminated
{1} ok boot disk

Netra 440, No Keyboard
Copyright 1998-2003 Sun Microsystems, Inc.  All rights reserved.
OpenBoot 4.10.3, 4096 MB memory installed, Serial #53005571.
Ethernet address 0:3:ba:28:cd:3, Host ID: 8328cd03.

Initializing     1MB of memory at addr        123fecc000 -

Initializing     1MB of memory at addr        123fe02000 -

Initializing    14MB of memory at addr        123f002000 -

Initializing    16MB of memory at addr        123e002000 -

Initializing   992MB of memory at addr        1200000000 -

Initializing  1024MB of memory at addr        1000000000 -

Initializing  1024MB of memory at addr         200000000 -

Initializing  1024MB of memory at addr                 0 -

Rebooting with command: boot disk
Boot device: /pci@1f,700000/scsi@2/disk@0,0  File and args:
\
SunOS Release 5.8 Version Generic_114696-04 64-bit
Copyright 1983-2003 Sun Microsystems, Inc.  All rights reserved.
Hardware watchdog enabled
Indicator SYS_FRONT.ACT is now ON
configuring IPv4 interfaces: ce0.
Hostname: Sun-SFV440-a
The system is coming up.  Please wait.
NIS domainname is Ecd.East.Sun.COM
Starting IPv4 router discovery.
starting rpc services: rpcbind keyserv ypbind done.
Setting netmask of lo0 to 255.0.0.0
Setting netmask of ce0 to 255.255.255.0
Setting default IPv4 interface for multicast: add net 224.0/4: gateway Sun-
SFV440-a
```

CODE EXAMPLE 4-10 `consolehistory run -v` Command Output *(Continued)*

```
syslog service starting.
Print services started.
volume management starting.
The system is ready.

Sun-SFV440-a console login: May  9 14:52:57 Sun-SFV440-a rmclomv: NOTICE:
keyswitch change event - state = UNKNOWN

May  9 14:52:57 Sun-SFV440-a rmclomv: Keyswitch Position has changed to Unknown
state.

May  9 14:52:58 Sun-SFV440-a rmclomv: NOTICE: keyswitch change event - state =
LOCKED

May  9 14:52:58 Sun-SFV440-a rmclomv: KeySwitch Position has changed to Locked
State.

May  9 14:53:00 Sun-SFV440-a rmclomv: NOTICE: keyswitch change event - state =
NORMAL

May  9 14:53:01 Sun-SFV440-a rmclomv: KeySwitch Position has changed to On State.

sc>
```

8. **Examine the ALOM boot log. Type:**

```
sc> consolehistory boot -v
```

The ALOM boot log contains boot messages from POST, OpenBoot firmware, and Solaris software from the host server's most recent reset.

The following sample output shows the boot messages from POST.

CODE EXAMPLE 4-11 `consolehistory boot -v` Command Output (Boot Messages From POST)

```
Keyswitch set to diagnostic position.
@(#)OBP 4.10.3 2003/05/02 20:25 Netra 440
Clearing TLBs
Power-On Reset
Executing Power On SelfTest

0>@(#) Sun Fire[TM] V440 POST 4.10.3 2003/05/04 22:08
       /export/work/staff/firmware_re/post/post-build-
4.10.3/Fiesta/system/integrated  (firmware_re)
0>Hard Powerup RST thru SW
0>CPUs present in system: 0 1
```

CODE EXAMPLE 4-11 `consolehistory boot -v` Command Output (Boot Messages From POST) *(Continued)*

```
0>OBP->POST Call with %o0=00000000.01012000.
0>Diag level set to MIN.
0>MFG scrpt mode set NORM
0>I/O port set to TTYA.
0>
0>Start selftest...
1>Print Mem Config
1>Caches : Icache is ON, Dcache is ON, Wcache is ON, Pcache is ON.
1>Memory interleave set to 0
1>      Bank 0 1024MB : 00000010.00000000 -> 00000010.40000000.
1>      Bank 2 1024MB : 00000012.00000000 -> 00000012.40000000.
0>Print Mem Config
0>Caches : Icache is ON, Dcache is ON, Wcache is ON, Pcache is ON.
0>Memory interleave set to 0
0>      Bank 0 1024MB : 00000000.00000000 -> 00000000.40000000.
0>      Bank 2 1024MB : 00000002.00000000 -> 00000002.40000000.
0>INFO:
0>      POST Passed all devices.
0>
0>POST: Return to OBP.
```

The following sample output shows the initialization of the OpenBoot PROM.

CODE EXAMPLE 4-12 `consolehistory boot -v` Command Output (OpenBoot PROM Initialization)

```
Keyswitch set to diagnostic position.
@(#)OBP 4.10.3 2003/05/02 20:25 Netra 440
Clearing TLBs
POST Results: Cpu 0000.0000.0000.0000
  %o0 = 0000.0000.0000.0000 %o1 = ffff.ffff.f00a.2b73 %o2 = ffff.ffff.ffff.ffff
POST Results: Cpu 0000.0000.0000.0001
  %o0 = 0000.0000.0000.0000 %o1 = ffff.ffff.f00a.2b73 %o2 = ffff.ffff.ffff.ffff
Membase: 0000.0000.0000.0000
MemSize: 0000.0000.0004.0000
Init CPU arrays Done
Probing /pci@1d,700000 Device 1  Nothing there
Probing /pci@1d,700000 Device 2  Nothing there
```

The following sample output shows the system banner.

CODE EXAMPLE 4-13 `consolehistory boot -v` Command Output (System Banner Display)

```
Netra 440, No Keyboard
Copyright 1998-2003 Sun Microsystems, Inc.  All rights reserved.
OpenBoot 4.10.3, 4096 MB memory installed, Serial #53005571.
Ethernet address 0:3:ba:28:cd:3, Host ID: 8328cd03.
```

The following sample output shows OpenBoot Diagnostics testing.

CODE EXAMPLE 4-14 `consolehistory boot -v` Command Output (OpenBoot Diagnostics Testing)

```
Running diagnostic script obdiag/normal

Testing /pci@1f,700000/network@1
Testing /pci@1e,600000/ide@d
Testing /pci@1e,600000/isa@7/flashprom@2,0
Testing /pci@1e,600000/isa@7/serial@0,2e8
Testing /pci@1e,600000/isa@7/serial@0,3f8
Testing /pci@1e,600000/isa@7/rtc@0,70
Testing /pci@1e,600000/isa@7/i2c@0,320:tests=
{gpio@0.42,gpio@0.44,gpio@0.46,gpio@0.48}
Testing /pci@1e,600000/isa@7/i2c@0,320:tests={hardware-monitor@0.5c}
Testing /pci@1e,600000/isa@7/i2c@0,320:tests={temperature-sensor@0.9c}
Testing /pci@1c,600000/network@2
Testing /pci@1f,700000/scsi@2,1
Testing /pci@1f,700000/scsi@2
```

The following sample output shows memory initialization by the OpenBoot PROM.

CODE EXAMPLE 4-15 `consolehistory boot -v` Command Output (Memory Initialization)

```
Initializing     1MB of memory at addr        123fe02000 -

Initializing    12MB of memory at addr        123f000000 -

Initializing  1008MB of memory at addr        1200000000 -

Initializing  1024MB of memory at addr        1000000000 -

Initializing  1024MB of memory at addr         200000000 -

Initializing  1024MB of memory at addr                 0 -

{1} ok boot disk
```

The following sample output shows the system booting and loading Solaris software.

CODE EXAMPLE 4-16 `consolehistory boot -v` Command Output (System Booting and Loading Solaris Software)

```
Rebooting with command: boot disk
Boot device: /pci@1f,700000/scsi@2/disk@0,0  File and args:
Loading ufs-file-system package 1.4 04 Aug 1995 13:02:54.
FCode UFS Reader 1.11 97/07/10 16:19:15.
Loading: /platform/SUNW,Sun-Fire-V440/ufsboot
Loading: /platform/sun4u/ufsboot
\
SunOS Release 5.8 Version Generic_114696-04 64-bit
Copyright 1983-2003 Sun Microsystems, Inc.  All rights reserved.
Hardware watchdog enabled
sc>
```

9. **Type the** `showusers` **command.**

```
sc> showusers
```

This command displays all the users currently logged in to ALOM.

CODE EXAMPLE 4-17 ALOM Reports on Active User Sessions

```
username        connection      login time      client IP addr
console
--------         ----------       ----------       --------------
admin            serial           FEB 28 19:45     system
admin           net-1            MAR 03 14:43     129.111.111.111
sc>
```

In this case, notice that there are two separate simultaneous administrative users. The first is logged in through the SERIAL MGT port and has access to the system console. The second user is logged in through `telnet` connection from another host to the NET MGT port. The second user can view the system console session but cannot input console commands.

10. Type the `showplatform` command.

```
sc> showplatform
```

This command displays the status of the operating system, which may be Running, Stopped, Initializing, or in a handful of other states.

CODE EXAMPLE 4-18 ALOM Reports on Operating System Status

```
SUNW,Netra-440

Domain         Status
------         ------
vsp75-202-priv OS Running
```

11. Use ALOM to run POST diagnostics.

Doing this involves several steps.

a. Type:

```
sc> bootmode diag
```

This command temporarily overrides the server's OpenBoot Diagnostics `diag-switch?` setting, forcing power-on self-test (POST) diagnostics to run when power is cycled off and on. If the server is not power cycled within 10 minutes, it reverts back to its defaults.

b. Power cycle the system. Type:

```
sc> poweroff

Are you sure you want to power off the system [y/n]? y

sc> poweron
```

POST diagnostics begin to run as the system reboots. However, you will see no messages until you switch from ALOM to the system console. For details, refer to the *Netra 440 Server System Administration Guide*.

c. Switch to the system console. Type:

```
sc> console
Enter #. to return to ALOM.

0>@(#) Sun Fire[TM] V440 POST 4.10.0 2003/04/01 22:28

/export/work/staff/firmware_re/post/post-build-
4.10.0/Fiesta/system/integrated  (firmware_re)
0>Hard Powerup RST thru SW
0>CPUs present in system: 0 1 2 3
0>OBP->POST Call with %o0=00000000.01008000.
```

You should begin seeing console output and POST messages. The exact text that appears on your screen depends on the state of your Netra 440 server, and on how long you delay between powering on the system and switching to the system console.

Note – Any system console or POST messages you might miss are preserved in the ALOM boot log. To access the boot log, type **`consolehistory boot -v`** from the `sc>` prompt.

For more information about ALOM command-line functions, refer to the *Advanced Lights Out Manager User's Guide.*

For more information about controlling POST diagnostics, see "Controlling POST Diagnostics" on page 13.

For information about interpreting POST error messages, see "What POST Error Messages Tell You" on page 11.

Using Solaris System Information Commands

This section explains how to run Solaris system information commands on a Netra 440 server. To find out what these commands tell you, see "Solaris System Information Commands" on page 24, or see the appropriate man pages.

▼ To Use Solaris System Information Commands

1. **Decide what kind of system information you want to display.**

   For more information, see "Solaris System Information Commands" on page 24.

2. **Type the appropriate command at a system console prompt. See TABLE 4-1.**

TABLE 4-1 Using Solaris System Information Commands

Command	What It Displays	What to Type	Notes
`prtconf`	System configuration information	`/usr/sbin/prtconf`	—
`prtdiag`	Diagnostic and configuration information	`/usr/platform/‘uname -i‘/sbin/prtdiag`	Use the `-v` option for additional detail.
`prtfru`	FRU hierarchy and SEEPROM memory contents	`/usr/sbin/prtfru`	Use the `-l` option to display hierarchy. Use the `-c` option to display SEEPROM data.
`psrinfo`	Date and time each CPU came online; processor clock speed	`/usr/sbin/psrinfo`	Use the `-v` option to obtain clock speed and other data.
`showrev`	Hardware and software revision information	`/usr/bin/showrev`	Use the `-p` option to show software patches.

Using OpenBoot Information Commands

This section explains how to run OpenBoot commands that display different kinds of system information about a Netra 440 server. To find out what these commands tell you, see "Other OpenBoot Commands" on page 21, or refer to the appropriate man pages.

As long as you can get to the `ok` prompt, you can use OpenBoot information commands. This means the commands are usually accessible even when your system cannot boot its operating system software.

▼ To Use OpenBoot Information Commands

1. **If necessary, shut down the system to reach the `ok` prompt.**

   How you do this depends on the system's condition. If possible, you should warn users and shut down the system gracefully. For information, refer to the *Netra 440 Server System Administration Guide*.

2. **Decide what kind of system information you want to display.**

   For more information, see "Other OpenBoot Commands" on page 21.

3. **Type the appropriate command at a system console prompt. See TABLE 4-2.**

TABLE 4-2 Using OpenBoot Information Commands

Command to Type	What It Displays
`printenv`	OpenBoot configuration variable defaults and settings

`probe-scsi`
`probe-scsi-all`
`probe-ide`
	Target address, unit number, device type, and manufacturer name of active SCSI and IDE devices

`show-devs`	Hardware device paths of all devices in the system configuration

CHAPTER 5

Exercising the System

Sometimes a server exhibits a problem that cannot be isolated definitively to a particular hardware or software component. In such cases, it may be useful to run a diagnostic tool that stresses the system by continuously running a comprehensive battery of tests. Sun provides the SunVTS software that you can use with the Netra 440 server.

This chapter describes the tasks necessary to use SunVTS software to exercise your Netra 440 server.

Tasks covered in this chapter include:

- "Exercising the System Using SunVTS Software" on page 86
- "Checking Whether SunVTS Software Is Installed" on page 90

If you want *background information* about the tools and when to use them, turn to Chapter 1 and Chapter 2.

Note – The procedures in this chapter assume that you are familiar with the OpenBoot firmware and that you know how to access the `ok` prompt. For background information and instructions, refer to the *Netra 440 Server System Administration Guide*.

Exercising the System Using SunVTS Software

The SunVTS 5.1 Patch Set 4 (PS4) software, and future compatible versions, are supported on the Netra 440 server. You can download the most recent SunVTS software from `http://www.sun.com/oem/products/vts/`.

Before you begin, the Solaris OS must be running. You also need to ensure that SunVTS validation test software is installed on your system. See "Checking Whether SunVTS Software Is Installed" on page 90.

SunVTS software requires that you use one of two security schemes. The security scheme you choose must be properly configured in order for you to perform this procedure. For details, see:

- *SunVTS User's Guide*
- "SunVTS Software and Security" on page 38

SunVTS software features both character-based and graphics-based interfaces. This procedure assumes that you are using the graphical user interface (GUI) on a system running the Common Desktop Environment (CDE). For more information about the character-based SunVTS TTY interface, and specifically for instructions on accessing it bby TIP or `telnet` commands, refer to the *SunVTS User's Guide.*

SunVTS software can be run in several modes. This procedure assumes that you are using the default Functional mode. For a synopsis of the modes, see "Exercising the System Using SunVTS Software" on page 37.

This procedure also assumes that the Netra 440 server is "headless"—that is, it is not equipped with a monitor capable of displaying bitmapped graphics. In this case, you access the SunVTS GUI by logging in remotely from a machine that has a graphics display.

Finally, this procedure describes how to run SunVTS tests in general. Individual tests may presume the presence of specific hardware, or may require specific drivers, cables, or loopback connectors. For information about test options and prerequisites, refer to:

- *SunVTS Test Reference Manual*
- *SunVTS Documentation Supplement*

To Exercise the System Using SunVTS Software

1. **Log in as superuser to a system with a graphics display.**

   The display system should be one with a frame buffer and monitor capable of displaying bitmapped graphics such as those produced by the SunVTS GUI.

2. **Enable remote display. On the display system, type:**

   ```
   # /usr/openwin/bin/xhost + test-system
   ```

   where *test-system* is the name of the Netra 440 server being tested.

3. **Remotely log in to the Netra 440 server as superuser.**

   Use a command such as `rlogin` or `telnet`.

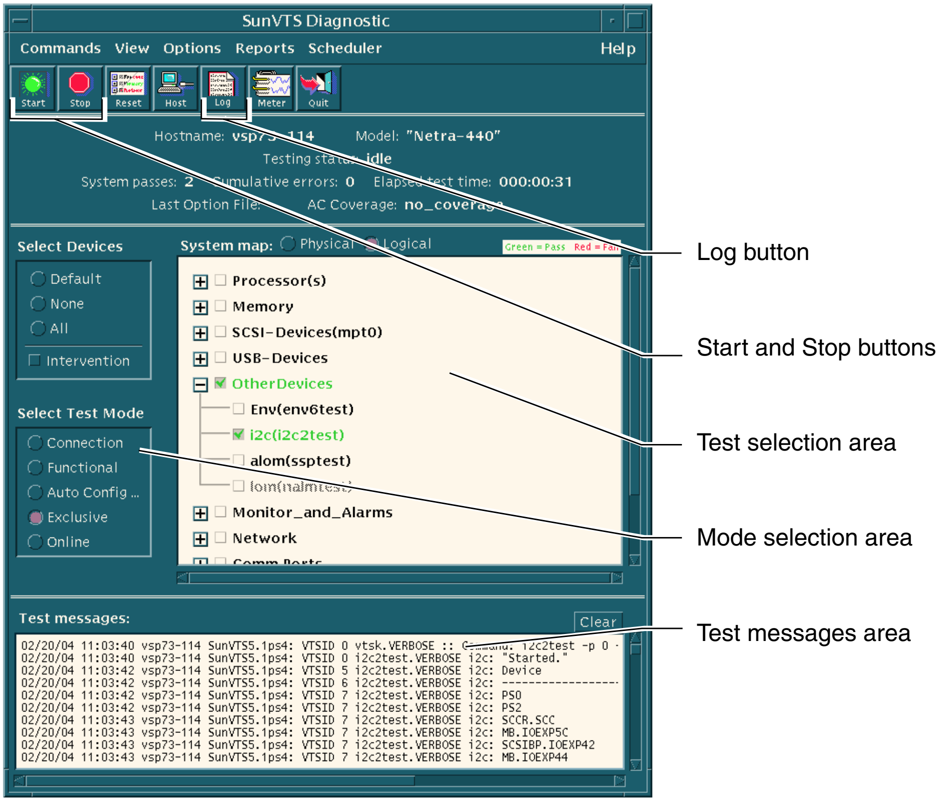
4. **Start SunVTS software. Type:**

   ```
   # /opt/SUNWvts/bin/sunvts -display display-system:0
   ```

   where *display-system* is the name of the machine through which you are remotely logged in to the Netra 440 server.

   If you have installed SunVTS software in a location other than the default `/opt` directory, alter the path in the above command accordingly.

The SunVTS GUI appears on the display system's screen.



Hostname: vsp73-114 Model: "Netra-440"

Testing status: idle

System passes: 2 Cumulative errors: 0 Elapsed test time: 000:00:31

Last Option File: AC Coverage: no_coverage

Select Devices

- ○ Default
- ○ None
- ○ All
- □ Intervention

System map: ○ Physical ● Logical

- ⊞ □ Processor(s)
- ⊞ □ Memory
- ⊞ □ SCSI-Devices(mpt0)
- ⊞ □ USB-Devices
- ⊟ ☑ OtherDevices
  - □ Env(env6test)
  - ☑ i2c(i2c2test)
  - □ alom(ssptest)
  - □ lom(nalmtest)
- ⊞ □ Monitor_and_Alarms
- ⊞ □ Network
- ⊞ □ Comm Ports

Select Test Mode

- ○ Connection
- ○ Functional
- ○ Auto Config ...
- ● Exclusive
- ○ Online

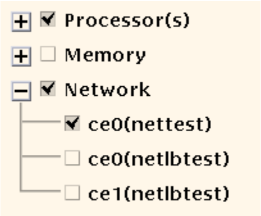
Test messages:

```
02/20/04 11:03:40 vsp73-114 SunVTS5.1ps4: VTSID 0 vtsk.VERBOSE :: Command: i2c2test -p 0 -
02/20/04 11:03:40 vsp73-114 SunVTS5.1ps4: VTSID 0 i2c2test.VERBOSE i2c: "Started."
02/20/04 11:03:42 vsp73-114 SunVTS5.1ps4: VTSID 5 i2c2test.VERBOSE i2c: Device
02/20/04 11:03:42 vsp73-114 SunVTS5.1ps4: VTSID 6 i2c2test.VERBOSE i2c: ------------------
02/20/04 11:03:42 vsp73-114 SunVTS5.1ps4: VTSID 7 i2c2test.VERBOSE i2c: PS0
02/20/04 11:03:42 vsp73-114 SunVTS5.1ps4: VTSID 7 i2c2test.VERBOSE i2c: PS2
02/20/04 11:03:43 vsp73-114 SunVTS5.1ps4: VTSID 7 i2c2test.VERBOSE i2c: SCCR.SCC
02/20/04 11:03:43 vsp73-114 SunVTS5.1ps4: VTSID 7 i2c2test.VERBOSE i2c: MB.IOEXP5C
02/20/04 11:03:43 vsp73-114 SunVTS5.1ps4: VTSID 7 i2c2test.VERBOSE i2c: SCSIBP.IOEXP42
02/20/04 11:03:43 vsp73-114 SunVTS5.1ps4: VTSID 7 i2c2test.VERBOSE i2c: MB.IOEXP44
```

FIGURE 5-1 The SunVTS GUI Screen

5. **Expand the test lists to see the individual tests.**

   The interface's test selection area lists tests in categories, such as "Network," as shown below. To expand a category, right-click the ⊞ icon to the left of the category name.



- ⊞ ☑ Processor(s)
- ⊞ □ Memory
- ⊟ ☑ Network
  - ☑ ce0(nettest)
  - □ ce0(netlbtest)
  - □ ce1(netlbtest)

6. **(*Optional*) Select the tests you want to run.**

   Certain tests are enabled by default, and you can choose to accept these.

   Alternatively, you can enable and disable individual tests or blocks of tests by clicking the checkbox next to the test name or test category name. Tests are enabled when checked, and disabled when not checked. TABLE 5-1 lists tests that are especially useful to run on a Netra 440 server.

Note – TABLE 5-1 lists FRUs in order of the likelihood they caused the test to fail.

---

7. **(*Optional*) Customize individual tests.**

   You can customize individual tests by right-clicking on the name of the test. For instance, in the illustration under Step 5, right-clicking on the text string `ce0(nettest)` brings up a menu that lets you configure this Ethernet test.

TABLE 5-1 Useful SunVTS Tests to Run on a Netra 440 Server

SunVTS Tests	FRUs Exercised by Tests
`cputest`, `fputest`, `iutest`, `l1dcachetest` — *indirectly*: `l2cachetest`, `l2sramtest`, `mptest`, `mpconstest`, `systest`	CPU/memory module, motherboard
`disktest`	Disks, cables, SCSI backplane
`dvdtest`, `cdtest`	DVD device, cable, motherboard
`env6test`, `i2c2test`	Power supplies, fan trays, LEDs, motherboard, ALOM card, system configuration card (SCC), CPU/memory module, DIMMs, SCSI backplane
`nettest`, `netlbtest`	Network interface, network cable, motherboard
`pmemtest`, `vmemtest`	DIMMs, CPU/memory module, motherboard
`ssptest`	ALOM card
`sutest`	Motherboard (serial port `ttyb`)
`usbkbtest`, `disktest`	USB devices, cable, motherboard (USB controller)
`nalmtest`	Alarm card test
`ramtest`	Memory test
`bustest`	System bus test

8. **Start testing.**

   Click the Start button, located at the top left of the SunVTS window, to begin running the tests you enabled. Status and error messages appear in the test messages area located across the bottom of the window. You can stop testing at any time by clicking the Stop button.

   During testing, SunVTS software logs all status and error messages. To view these, click the Log button or select Log Files from the Reports menu. This opens a log window from which you can choose to view the following logs:

   - *Information* – Detailed versions of all the status and error messages that appear in the test messages area.
   - *Test Error* – Detailed error messages from individual tests.
   - *VTS Kernel Error* – Error messages pertaining to SunVTS software itself. You should look here if SunVTS software appears to be acting strangely, especially when it starts up.
   - *UNIX Messages* (`/var/adm/messages`) – A file containing messages generated by the operating system and various applications.
   - *Log Files* (`/var/opt/SUNWvts/logs`) – A directory containing the log files.

   For further information, refer to the manuals that accompany SunVTS software. These are listed in the section "Related Documentation" on page xiv.

Checking Whether SunVTS Software Is Installed

SunVTS software consists of optional packages that may or may not have been loaded when your system software was installed.

In addition to the SunVTS packages themselves, SunVTS software starting with version 5.1 requires certain XML and run-time library packages that may not be installed by default on Solaris software.

This procedure assumes that the Solaris OS is running on the Netra 440 server, and that you have access to the Solaris command line. For more information, refer to the *Netra 440 Server System Administration Guide*.

▼ To Check Whether SunVTS Software Is Installed

1. **Check for the presence of SunVTS packages. Type:**

```
% pkginfo -l SUNWvts SUNWvtsx SUNWvtsmn
```

- If SunVTS software is loaded, information about the packages is displayed.
- If SunVTS software is not loaded, you see an error message for each missing package.

```
ERROR: information for "SUNWvts" was not found
ERROR: information for "SUNWvtsx" was not found
...
```

The pertinent packages are as follows.

Package	Description
SUNWvts	SunVTS kernel, user interface, and 32-bit binary tests
SUNWvtsx	SunVTS 64-bit binary tests and kernel
SUNWvtsmn	SunVTS man pages

2. **(*Solaris 8 only*) Check for additional needed software.**

This applies only if you intend to install and run SunVTS 5.1 software (or later compatible versions) under Solaris 8.

SunVTS 5.1 software requires additional packages that may not be installed with Solaris 8 software. To find out, type the following:

```
% pkginfo -l SUNWlxml SUNWlxmlx SUNWzlib SUNWzlibx
```

This tests for the presence of the following packages.

Package	Description	Notes
SUNXlxml	XML library (32-bit)	Required by SunVTS 5.1
SUNWlxmlx	XML library (64-bit)	
SUNWzlib	Zip compression library (32-bit)	Needed by XML libraries
SUNWzlibx	Zip compression library (64-bit)	

3. **If necessary, load any missing packages.**

   Use the pkgadd utility to load onto your system any SunVTS and support packages that you determined you needed in Step 1 or Step 2.

   For Solaris 8, the SunVTS and XML packages are included on the Software Supplement CD. The zlib packages are included on the Solaris primary installation CD in the Entire Solaris Software Group.

   Note that /opt/SUNWvts is the default directory for installing SunVTS software.

4. **Load SunVTS patches, if appropriate.**

   Patches to SunVTS software are available periodically on the SunSolve Online^{SM} Web site. These patches provide enhancements and bug fixes. In some cases, there are tests that will not run properly unless the patches are installed.

   For installation information, refer to the *SunVTS User's Guide*, the appropriate Solaris documentation, and the pkgadd man page.

PART II Troubleshooting

The following chapters within this part of the *Netra 440 Server Diagnostics and Troubleshooting Guide* provide you with approaches for avoiding and troubleshooting problems that might arise from hardware defects.

For background information about diagnostic tools, as well as detailed instructions on how to use the tools, see the chapters in Part I – Diagnostics.

Chapters included in Part II are:

- Chapter 6 – Troubleshooting Options
- Chapter 7 – Troubleshooting Hardware Problems

CHAPTER 6

Troubleshooting Options

There are several troubleshooting options that you can implement when you set up and configure the Netra 440 server. By setting up your system with troubleshooting in mind, you can save time and minimize disruptions if the system encounters any problems.

Tasks covered in this chapter include:

- "To Enable the Core Dump Process" on page 103
- "Testing the Core Dump Setup" on page 105

Other information in this chapter includes:

- "Updated Troubleshooting Information" on page 95
- "Firmware and Software Patch Management" on page 97
- "Sun Install Check Tool" on page 97
- "Sun Explorer Data Collector" on page 98
- "Configuring the System for Troubleshooting" on page 99

Updated Troubleshooting Information

Sun will continue to gather and publish information about the Netra 440 server long after the initial system documentation is shipped. You can obtain the most current server troubleshooting information in the Product Notes and at Sun web sites. These resources can help you understand and diagnose problems that you might encounter.

ReleaseNotes

Netra 440 Server Release Notes (817-3885-xx) contain late-breaking information about the system, including the following:

- Current recommended and required software patches
- Updated hardware and driver compatibility information
- Known issues and bug descriptions, including solutions and workarounds

The latest Release Notes are available at:

`http://www.sun.com/documentation`

Web Sites

SunSolve Online

This site presents a collection of resources for Sun technical and support information. Access to some of the information on this site depends on the level of your service contract with Sun. This site includes the following:

- *Patch Support Portal* – Everything you need to download and install patches, including tools, product patches, security patches, signed patches, x86 drivers, and more.
- *Sun Install Check tool* – A utility you can use to verify proper installation and configuration of a new Netra server. This resource checks a Netra server for valid patches, hardware, operating environment, and configuration.
- *Sun System Handbook* – A document that contains technical information and provides access to discussion groups for most Sun hardware, including the Netra 440 server.
- Support documents, security bulletins, and related links.

The SunSolve Online Web site is at:

`http://sunsolve.sun.com`

Big Admin

This web site is a one-stop resource for Sun system administrators. The Big Admin web site is at:

`http://www.sun.com/bigadmin`

Firmware and Software Patch Management

Sun makes every attempt to ensure that each system is shipped with the latest firmware and software. However, in complex systems, bugs and problems are discovered in the field after systems leave the factory. Often, these problems are fixed with patches to the system's firmware. Keeping your system's firmware and Solaris OS current with the latest recommended and required patches can help you avoid problems that others might have already discovered and solved.

Firmware and operating system updates are often required to diagnose or fix a problem. Schedule regular updates of your system's firmware and software so that you will not have to update the firmware or software at an inconvenient time.

You can find the latest patches and updates for the Netra 440 server at the Web sites listed in "Web Sites" on page 96.

Sun Install Check Tool

When you install the Sun^{SM} Install Check tool, you also install Sun Explorer Data Collector. The Sun Install Check tool uses Sun Explorer Data Collector to help you confirm that Netra 440 server installation has been completed optimally. Together, they can evaluate your system for the following:

- Minimum required operating system level
- Presence of key critical patches
- Proper system firmware levels
- Unsupported hardware components

If potential issues are identified, the software generates a report that will provide specific instructions to remedy the issues.

You can download the Sun Install Check tool software and documentation at:

`http://www.sun.com/software/installcheck/`

Sun Explorer Data Collector

The Sun Explorer Data Collector is a system data collection tool that Sun support services engineers sometimes use when troubleshooting Sun SPARC and x86 systems. In certain support situations, Sun support services engineers might ask you to install and run this tool. If you installed the Sun Install Check tool at initial installation, you also installed Sun Explorer Data Collector. If you did not install the Sun Install Check tool, you can install Sun Explorer Data Collector later without the Sun Install Check tool. By installing this tool as part of your initial system setup, you avoid having to install the tool at a later, and often inconvenient time.

Both the Sun Install Check tool (with bundled Sun Explorer Data Collector) and the Sun Explorer Data Collector (standalone) are available at:

http://sunsolve.sun.com

At that site, click on the appropriate link.

Sun Remote Services Net Connect

Sun^{SM} Remote Services (SRS) Net Connect is a collection of system management services designed to help you better control your computing environment. These Web-delivered services enable you to monitor systems, to create performance and trend reports, and to receive automatic notification of system events. These services help you to act more quickly when a system event occurs and to manage potential issues before they become problems.

More information about SRS Net Connect is available at:

http://www.sun.com/service/support/srs/netconnect

Configuring the System for Troubleshooting

System failures are characterized by certain symptoms. Each symptom can be traced to one or more problems or causes by using specific troubleshooting tools and techniques. This section describes troubleshooting tools and techniques that you can control through configuration variables.

Hardware Watchdog Mechanism

The hardware watchdog mechanism is a hardware timer that is continually reset as long as the operating system is running. If the system hangs, the operating system is no longer able to reset the timer. The timer then expires and causes an automatic externally initiated reset (XIR), displaying debug information on the system console. The hardware watchdog mechanism is enabled by default. If the hardware watchdog mechanism is disabled, the Solaris OS must be configured before the hardware watchdog mechanism can be reenabled.

The configuration variable `error-reset-recovery` allows you to control how the hardware watchdog mechanism behaves when the timer expires. The following are the error-reset-recovery settings:

- `boot` (default) – Resets the timer and attempts to reboot the system
- `sync` (recommended) – Attempts to automatically generate a core dump file dump, reset the timer, and reboot the system
- `none` (equivalent to issuing a manual XIR from the ALOM system controller**)** – Drops the server to the `ok` prompt, enabling you to issue commands and debug the system

For more information about the hardware watchdog mechanism and XIR, refer to the *Netra 440 Server System Administration Guide* (817-3884-xx).

For information about troubleshooting system hangs, see:

- "Responding to System Hang States" on page 111
- "Troubleshooting a System That Is Hanging" on page 147

Automatic System Recovery Settings

The automatic system recovery (ASR) features enable the system to resume operation after experiencing certain nonfatal hardware faults or failures. When ASR is enabled, the system's firmware diagnostics automatically detect failed hardware components. An auto-configuring capability designed into the OpenBoot firmware enables the system to unconfigure failed components and to restore system operation. As long as the system is capable of operating without the failed component, the ASR features enable the system to reboot automatically, without operator intervention.

How you configure ASR settings effects not only how the system handles certain types of failures but also on how you go about troubleshooting certain problems.

For day-to-day operations, enable ASR by setting OpenBoot configuration variables as shown in TABLE 6-1.

TABLE 6-1 OpenBoot Configuration Variable Settings to Enable Automatic System Recovery

Variable	Setting
`auto-boot?`	`true`
`auto-boot-on-error?`	`true`
`diag-level`	`max`
`diag-switch?`	`true`
`diag-trigger`	`all-resets`
`post-trigger`	`all-resets`
`diag-device`	(Set to the boot-device value)

Configuring your system this way ensures that diagnostic tests run automatically when most serious hardware and software errors occur. With this ASR configuration, you can save time diagnosing problems since POST and OpenBoot Diagnostics test results are already available after the system encounters an error.

For more information about how ASR works, and complete instructions for enabling ASR capability, refer to the *Netra 440 Server System Administration Guide* (817-3884-xx).

Remote Troubleshooting Capabilities

You can use the Advanced Lights Out Manager (ALOM) system controller to troubleshoot and diagnose the system remotely. The ALOM system controller lets you do the following:

- Turn system power on and off
- Control the Locator LED
- Change OpenBoot configuration variables
- View system environmental status information
- View system event logs

In addition, you can use the ALOM system controller to access the system console, provided it has not been redirected. System console access enables you to do the following:

- Run OpenBoot Diagnostics tests
- View Solaris OS output
- View POST output
- Issue firmware commands at the `ok` prompt
- View error events when the Solaris OS terminates abruptly

For more information about ALOM, see:

- "Monitoring the System Using Advanced Lights Out Manager" on page 35
- "Monitoring the System Using Sun Advanced Lights Out Manager" on page 68
- *Advanced Lights Out Manager Software User's Guide for the Netra 440 Server*

For more information about the system console, refer to the *Netra 440 Server System Administration Guide*.

System Console Logging

Console logging is the ability to collect and log system console output. Console logging captures console messages so that system failure data, like Fatal Reset error details and POST output, can be recorded and analyzed.

Console logging is especially valuable when troubleshooting Fatal Reset errors and RED State Exceptions. In these conditions, the Solaris OS terminates abruptly, and although it sends messages to the system console, the operating sysem software does not log any messages in traditional file system locations like the /var/adm/messages file. The following is an excerpt from the /var/adm/messages file.

CODE EXAMPLE 6-1 /var/adm/messages File Information

```
May  9 08:42:17 Sun-SFV440-a SUNW,UltraSPARC-IIIi: [ID 904467 kern.info] NOTICE:
[AFT0] Corrected memory (RCE) Event detected by CPU0 at TL=0, errID
0x0000005f.4f2b0814
May  9 08:42:17 Sun-SFV440-a     AFSR 0x00100000<PRIV>.82000000<RCE> AFAR
0x00000023.3f808960
May  9 08:42:17 Sun-SFV440-a     Fault_PC <unknown> J_REQ 2
May  9 08:42:17 Sun-SFV440-a     MB/P2/B0: J0601 J0602
May  9 08:42:17 Sun-SFV440-a unix: [ID 752700 kern.warning] WARNING: [AFT0]
Sticky Softerror encountered on Memory Module MB/P2/B0: J0601 J0602
May  9 08:42:19 Sun-SFV440-a SUNW,UltraSPARC-IIIi: [ID 263516 kern.info] NOTICE:
[AFT0] Corrected memory (CE) Event detected by CPU2 at TL=0, errID
0x0000005f.c52f509c
```

The error logging daemon, syslogd, automatically records various system warnings and errors in message files. By default, many of these system messages are displayed on the system console and are stored in the /var/adm/messages file. You can direct where these messages are stored or have them sent to a remote system by setting up system message logging. For more information, refer to “How to Customize System Message Logging” in the *System Administration Guide: Advanced Administration*, which is part of the Solaris System Administrator Collection.

In some failure situations, a large stream of data is sent to the system console. Because ALOM log messages are written into a “circular buffer” that holds 64 Kbyte of data, it is possible that the output identifying the original failing component can be overwritten. Therefore, you may want to explore further system console logging options, such as SRS Net Connect or third-party vendor solutions. For more information about SRS Net Connect, see “Sun Remote Services Net Connect” on page 98.

More information about SRS Net Connect is available at:

http://www.sun.com/service/support/

Certain third-party vendors offer data logging terminal servers and centralized system console management solutions that monitor and log output from many systems. Depending on the number of systems you are administering, these might offer solutions for logging system console information.

For more information about the system console, refer to the *Netra 440 Server System Administration Guide*.

The Core Dump Process

In some failure situations, a Sun engineer might need to analyze a system core dump file to determine the root cause of a system failure. Although the core dump process is enabled by default, you should configure your system so that the core dump file is saved in a location with adequate space. You might also want to change the default core dump directory to another locally mounted location so that you can better manage any system core dumps. In certain testing and preproduction environments, this is recommended since core dump files can take up a large amount of file system space.

Swap space is used to save the dump of system memory. By default, Solaris software uses the first swap device that is defined. This first swap device is known as the *dump device*.

During a system core dump, the system saves the content of kernel core memory to the dump device. The dump content is compressed during the dump process at a 3:1 ratio; that is, if the system were using 6 Gbyte of kernel memory, the dump file will be about 2 Gbyte. For a typical system, the dump device should be at least one–third the size of the total system memory.

See "To Enable the Core Dump Process" on page 103 for instructions on how to calculate the amount of available swap space. You would normally enable the core dump process just prior to placing a system into the production environment.

▼ To Enable the Core Dump Process

1. **Access the system console.**

   Refer to the *Netra 440 Server System Administration Guide*.

2. **Check that the core dump process is enabled.**

   As superuser, type the `dumpadm` command.

```
# dumpadm
Dump content: kernel pages
Dump device: /dev/dsk/c0t0d0s1 (swap)
Savecore directory: /var/crash/machinename
Savecore enabled: yes
```

By default, the core dump process is enabled in Solaris 8.

3. **Verify that there is sufficient swap space to dump memory.**

   Type the swap -l command.

```
# swap -l
swapfile             dev       swaplo     blocks     free
/dev/dsk/c0t3d0s0    32,24     16         4097312    4062048
/dev/dsk/c0t1d0s0    32,8      16         4097312    4060576
/dev/dsk/c0t1d0s1    32,9      16         4097312    4065808
```

   To determine how many bytes of swap space are available, multiply the number in the blocks column by 512. Taking the number of blocks from the first entry, c0t3d0s0, calculate as follows:

   4097312 x 512 = 2097823744

   The result is approximately 2 Gbyte.

4. **Verify that there is sufficient file system space for the core dump files.**

   Type the df -k command.

```
# df -k /var/crash/`uname -n`
```

   By default the location where savecore files are stored is:

   /var/crash/`uname -n`

   For instance, for the *mysystem* server, the default directory is:

   /var/crash/*mysystem*

   The file system specified must have space for the core dump files.

   If you see messages from savecore indicating not enough space in the /var/crash/ file, any other locally mounted (not NFS) file system can be used. Following is a sample message from savecore.

```
System dump time: Wed Apr 23 17:03:48 2003
savecore: not enough space in /var/crash/sf440-a (216 MB avail,
246 MB needed)
```

   Perform Step 5 and Step 6 if there is not enough space.

5. **Type the `df -k1` command to identify locations with more space.**

```
# df -k1
Filesystem         kbytes    used    avail capacity Mounted on
/dev/dsk/c1t0d0s0   832109    552314  221548   72%  /
/proc                     0       0       0     0%  /proc
fd                        0       0       0     0%  /dev/fd
mnttab                    0       0       0     0%  /etc/mntab
swap               3626264      16  362624     81%  /var/run
swap               3626656     408  362624     81%  /tmp
/dev/dsk/c1t0d0s7  33912732      9 33573596     1%  /export/home
```

6. **Type the `dumpadm -s` command to specify a location for the dump file.**

```
# dumpadm -s /export/home/
      Dump content: kernel pages
        Dump device: /dev/dsk/c3t5d0s1 (swap)
Savecore directory: /export/home
  Savecore enabled: yes
```

The `dumpadm -s` command enables you to specify the location for the swap file. See the `dumpadm` (1M) man page for more information.

Testing the Core Dump Setup

Before placing the system into a production environment, it might be useful to test whether the core dump setup works. This procedure might take some time depending on the amount of installed memory.

▼ To Test the Core Dump Setup

1. **Back up all your data and access the system console.**

2. **Gracefully shut down the system using the `shutdown` command.**

3. **At the `ok` prompt, issue the `sync` command.**

   You should see “dumping” messages on the system console.

   The system reboots. During this process, you can see the `savecore` messages.

4. **Wait for the system to finish rebooting.**

5. **Look for system core dump files in your** `savecore` **directory.**

   The files are named `unix.`*y* and `vmcore.`*y*, where *y* is the integer dump number. There should also be a `bounds` file that contains the next crash number `savecore` will use.

   If a core dump is not generated, perform the procedure described in "To Enable the Core Dump Process" on page 103.

CHAPTER 7

Troubleshooting Hardware Problems

The term *troubleshooting* refers to the act of applying diagnostic tools—often heuristically and accompanied by common sense—to determine the causes of system problems.

Each system problem must be treated on its own merits. It is not possible to provide a cookbook of actions that resolve each problem. However, this chapter provides some approaches and procedures, which used in combination with experience and common sense, can resolve many problems that might arise.

Tasks covered in this chapter include:

- "Troubleshooting a System With the Operating System Responding" on page 114
- "Troubleshooting a System After an Unexpected Reboot" on page 119
- "Troubleshooting Fatal Reset Errors and RED State Exceptions" on page 130
- "Troubleshooting a System That Does Not Boot" on page 141
- "Troubleshooting a System That Is Hanging" on page 147

Other information in this chapter includes:

- "Information to Gather During Troubleshooting" on page 108
- "System Error States" on page 111
- "Unexpected Reboots" on page 114

Information to Gather During Troubleshooting

Familiarity with a wide variety of equipment, and experience with a particular machine's common failure modes can be invaluable when troubleshooting system problems. Establishing a systematic approach to investigating and solving a particular system's problems can help ensure that you can quickly identify and remedy most issues as they arise.

The Netra 440 server indicates and logs events and errors in a variety of ways. Depending on the system's configuration and software, certain types of errors are captured only temporarily. Therefore, you must observe and record all available information immediately before you attempt any corrective action. POST, for instance, accumulates a list of failed components across resets. However, failed component information is cleared after a system reset. Similarly, the state of LEDs in a hung system is lost when the system reboots or resets.

If you encounter any system problems that are not familiar to you, gather as much information as you can before you attempt any remedial actions. The following task listing outlines a basic approach to information gathering.

- Gather as much error information (error indications and messages) as you can *from* the system. See "Error Information From the ALOM System Controller" on page 109 and "Error Information From the System" on page 109 for more information about sources of error indications and messages.
- Gather as much information as you can *about* the system by reviewing and verifying the system's operating system, firmware, and hardware configuration. To accurately analyze error indications and messages, you or a Sun support services engineer must know the system's operating system and patch revision levels as well as the specific hardware configuration. See "Recording Information About the System" on page 110.
- Compare the specifics of your situation to the latest published information about your system. Often, unfamiliar problems you encounter have been seen, diagnosed, and fixed by others. This information might help you avoid the unnecessary expense of replacing parts that are not actually failing. See "Updated Troubleshooting Information" on page 95 for information sources.

Error Information From the ALOM System Controller

In most troubleshooting situations, you can use the ALOM system controller as the primary source of information about the system. On the Netra 440 server, the ALOM system controller provides you with access to a variety of system logs and other information about the system, even when the system is powered off. For more information about ALOM, see:

- "Monitoring the System Using Advanced Lights Out Manager" on page 35
- "Monitoring the System Using Sun Advanced Lights Out Manager" on page 68
- *Advanced Lights Out Manager Software User's Guide for the Netra 440 Server*

Error Information From the System

Depending on the state of the system, you should check as many of the following sources as possible for error indications and record the information found.

- *Output from the* `prtdiag -v` *command* – If Solaris software is running, issue the `prtdiag -v` command to capture information stored by OpenBoot Diagnostics and POST tests. Any information from these tests about the current state of the system is lost when the system is reset. See "Troubleshooting a System With the Operating System Responding" on page 114.
- *Output from* `show-post-results` *and* `show-obdiag-results` *commands* – From the `ok` prompt, issue the `show-post-results` command or `show-obdiag-results` command to view summaries of the results from the most recent POST and OpenBoot Diagnostics tests, respectively. The test results are saved across power cycles and provide an indication of which components passed and which components failed POST or OpenBoot Diagnostics tests. See "Viewing Diagnostic Test Results After the Fact" on page 64.
- *State of system LEDs* – The system LEDs can be viewed in various locations on the system or by using the ALOM system controller. Be sure to check any network port LEDs for activity as you examine the system. Any information about the state of the system from the LEDs is lost when the system is reset. For more information about using LEDs to troubleshoot system problems, see "Isolating Faults Using LEDs" on page 57.
- *Solaris logs* – If Solaris software is running, check the message files in the `/var/adm/messages` file. For more information, refer to "How to Customize System Message Logging" in the *Solaris System Administration Guide: Advanced Administration Guide*, which is part of the Solaris System Administrator Collection.

- *System console* – You can access system console messages from OpenBoot Diagnostics and POST using the ALOM system controller, provided the system console has not been redirected. The system controller also provides you access to boot log information from the latest system reset. For more information about the system console, refer to the *Netra 440 Server System Administration Guide*.
- *Core files generated from panics* – These files are located in the `/var/crash` directory. See "The Core Dump Process" on page 103 for more information.

Recording Information About the System

As part of your standard operating procedures, it is important to have the following information about your system readily available:

- Current patch levels for the system firmware and operating system
- Solaris OS version
- Specific hardware configuration information
- Optional equipment and driver information
- Recent service records

Having all of this information available and verified makes it easier for you to recognize any problems already identified by others. This information is also required if you contact Sun support or your authorized support provider.

It is vital to know the version and patch revision levels of the system's operating system, patch revision levels of the firmware, and your specific hardware configuration before you attempt to fix any problems. Problems often occur after changes have been made to the system. Some errors are caused by hardware and software incompatibilities and interactions. If you have all system information available, you might be able to quickly fix a problem by simply updating the system's firmware. Knowing about recent upgrades or component replacements might help you avoid replacing components that are not faulty.

System Error States

When troubleshooting, it is important to understand what kind of error has occurred, to distinguish between real and apparent system hangs, and to respond appropriately to error conditions so as to preserve valuable information.

Responding to System Error States

Depending on the severity of a system error, a Netra 440 server might or might not respond to commands you issue to the system. Once you have gathered all available information, you can begin taking action. Your actions depend on the information you have already gathered and the state of the system.

Remember these guidelines:

- Avoid power cycling the system until you have gathered all the information you can. Error information might be lost when power cycling the system.
- If your system appears to be hung, attempt multiple approaches to get the system to respond. See "Responding to System Hang States" on page 111.

Responding to System Hang States

Troubleshooting a hanging system can be a difficult process because the root cause of the hang might be masked by false error indications from another part of the system. Therefore, it is important that you carefully examine all the information sources available to you before you attempt any remedy. Also, it is helpful to understand the type of hang the system is experiencing. This hang state information is especially important to Sun support services engineers, should you contact them.

A *system soft hang* can be characterized by any of the following symptoms:

- Usability or performance of the system gradually decreases.
- New attempts to access the system fail.
- Some parts of the system appear to stop responding.
- You can drop the system into the OpenBoot `ok` prompt level.

Some soft hangs might dissipate on their own, while others will require that the system be interrupted to gather information at the OpenBoot prompt level. A soft hang should respond to a break signal that is sent through the system console.

A *system hard hang* leaves the system unresponsive to a system break sequence. You will know that a system is in a hard hang state when you have attempted all the soft hang remedies with no success.

See "Troubleshooting a System That Is Hanging" on page 147.

Responding to Fatal Reset Errors and RED State Exceptions

Fatal Reset errors and RED State Exceptions are most often caused by hardware problems. Hardware Fatal Reset errors are the result of an "illegal" hardware state that is detected by the system. A hardware Fatal Reset error can either be a transient error or a hard error. A *transient error* causes intermittent failures. A *hard error* causes persistent failures that occur in the same way each time. CODE EXAMPLE 7-1 shows a sample Fatal Reset error alert from the system console.

CODE EXAMPLE 7-1 Fatal Reset Error Alert

```
Sun-SFV440-a console login:

Fatal Error Reset
CPU 0000.0000.0000.0002 AFSR 0210.9000.0200.0000  JETO PRIV OM TO
AFAR 0000.0280.0ec0.c180
SC Alert: Host System has Reset

SC Alert: Host System has read and cleared bootmode.
```

A RED State Exception condition is most commonly a hardware fault that is detected by the system. There is no recoverable information that you can use to troubleshoot a RED State Exception. The Exception causes a loss of system integrity, which would jeopardize the system if Solaris software continued to operate. Because of this, Solaris software terminates ungracefully without logging any details of the RED State Exception error in the `/var/adm/messages` file. CODE EXAMPLE 7-2 shows a sample RED State Exception alert from the system console.

CODE EXAMPLE 7-2 RED State Exception Alert

```
Sun-SFV440-a console login:

RED State Exception
Error enable reg: 0000.0001.00f0.001f
ECCR: 0000.0000.02f0.4c00
CPU: 0000.0000.0000.0002
TL=0000.0000.0000.0005  TT=0000.0000.0000.0010
   TPC=0000.0000.0100.4200  TnPC=0000.0000.0100.4204  TSTATE=
0000.0044.8200.1507
```

CODE EXAMPLE 7-2 RED State Exception Alert *(Continued)*

```
TL=0000.0000.0000.0004  TT=0000.0000.0000.0010
   TPC=0000.0000.0100.4200  TnPC=0000.0000.0100.4204  TSTATE=
0000.0044.8200.1507
TL=0000.0000.0000.0003  TT=0000.0000.0000.0010
   TPC=0000.0000.0100.4680  TnPC=0000.0000.0100.4684  TSTATE=
0000.0044.8200.1507
TL=0000.0000.0000.0002  TT=0000.0000.0000.0034
   TPC=0000.0000.0100.7164  TnPC=0000.0000.0100.7168  TSTATE=
0000.0044.8200.1507
TL=0000.0000.0000.0001  TT=0000.0000.0000.004e
   TPC=0000.0001.0001.fd24  TnPC=0000.0001.0001.fd28  TSTATE=
0000.0000.8200.1207

SC Alert: Host System has Reset

SC Alert: Host System has read and cleared bootmode.
```

In some isolated cases, software can cause a Fatal Reset error or RED State Exception. Typically, these are device driver problems that can be identified easily. You can obtain this information through SunSolve Online (see "Web Sites" on page 96), or by contacting Sun or the third-party driver vendor.

The most important pieces of information to gather when diagnosing a Fatal Reset error or RED State Exception are:

- System console output at the time of the error
- Recent service history of systems that encounter Fatal Reset errors or RED State Exceptions

Capturing system console indications and messages at the time of the error can help you isolate the true cause of the error. In some cases, the true cause of the original error might be masked by false error indications from another part of the system. For example, POST results (shown by the output from the `prtdiag` command) might indicate failed components, when, in fact, the "failed" components are not the actual cause of the Fatal Reset error. In most cases, a good component will actually report the Fatal Reset error.

By analyzing the system console output at the time of the error, you can avoid replacing components based on these false error indications. In addition, knowing the service history of a system experiencing transient errors can help you avoid repeatedly replacing "failed" components that do not fix the problem.

Unexpected Reboots

Sometimes, a system might reboot unexpectedly. In that case, ensure that the reboot was not caused by a panic. For example, L2-cache errors, which occur in user space (not kernel space), might cause Solaris software to log the L2-cache failure data and reboot the system. The information logged might be sufficient to troubleshoot and correct the problem. If the reboot was not caused by a panic, it might be caused by a Fatal Reset error or a RED State Exception. See "Troubleshooting Fatal Reset Errors and RED State Exceptions" on page 130.

Also, system ASR and POST settings can determine the system response to certain error conditions. If POST is not invoked during the reboot process, or if the system diagnostics level is not set to `max`, you might need to run system diagnostics at a higher level of coverage to determine the source of the reboot if the system message and system console files do not clearly indicate the source of the reboot.

Troubleshooting a System With the Operating System Responding

This procedure assumes that the system console is in its default configuration, so that you are able to switch between the system controller and the system console. Refer to the *Netra 440 Server System Administration Guide*.

▼ To Troubleshoot a System With the Operating System Running

1. **Log in to the system controller and access the `sc>` prompt.**

   For information, refer to the *Netra 440 Server System Administration Guide*.

2. **Examine the ALOM event log. Type:**

```
sc> showlogs
```

The ALOM event log shows system events such as reset events and LED indicator state changes that have occurred since the last system boot. CODE EXAMPLE 7-3 shows a sample event log, which indicates that the front panel Service Required LED is ON.

CODE EXAMPLE 7-3 `showlogs` Command Output

```
MAY 09 16:54:27 Sun-SFV440-a: 00060003: "SC System booted."
MAY 09 16:54:27 Sun-SFV440-a: 00040029: "Host system has shut down."
MAY 09 16:56:35 Sun-SFV440-a: 00060000: "SC Login: User admin Logged on."
MAY 09 16:56:54 Sun-SFV440-a: 00060000: "SC Login: User admin Logged on."
MAY 09 16:58:11 Sun-SFV440-a: 00040001: "SC Request to Power On Host."
MAY 09 16:58:11 Sun-SFV440-a: 00040002: "Host System has Reset"
MAY 09 16:58:13 Sun-SFV440-a: 0004000b: "Host System has read and cleared
                                         bootmode."
MAY 09 16:58:13 Sun-SFV440-a: 0004004f: "Indicator PS0.POK is now ON"
MAY 09 16:58:13 Sun-SFV440-a: 0004004f: "Indicator PS1.POK is now ON"
MAY 09 16:59:19 Sun-SFV440-a: 00040002: "Host System has Reset"
MAY 09 17:00:46 Sun-SFV440-a: 00040002: "Host System has Reset"
MAY 09 17:01:51 Sun-SFV440-a: 0004004f: "Indicator SYS_FRONT.SERVICE is now ON"
MAY 09 17:03:22 Sun-SFV440-a: 00040002: "Host System has Reset"
MAY 09 17:03:22 Sun-SFV440-a: 0004004f: "Indicator SYS_FRONT.SERVICE is now OFF"
MAY 09 17:03:24 Sun-SFV440-a: 0004000b: "Host System has read and cleared
                                         bootmode."
MAY 09 17:04:30 Sun-SFV440-a: 00040002: "Host System has Reset"
MAY 09 17:05:59 Sun-SFV440-a: 00040002: "Host System has Reset"
MAY 09 17:06:40 Sun-SFV440-a: 0004004f: "Indicator SYS_FRONT.SERVICE is now ON"
MAY 09 17:07:44 Sun-SFV440-a: 0004004f: "Indicator SYS_FRONT.ACT is now ON"
sc>
```

Note – Time stamps for ALOM logs reflect UTC (Universal Time Coordinated) time, while time stamps for the Solaris OS reflect local (server) time. Therefore, a single event might generate messages that appear to be logged at different times in different logs.

3. **Examine system environment status. Type:**

```
sc> showenvironment
```

The showenvironment command reports much useful data such as temperature readings; state of system and component LEDs; motherboard voltages; and status of system disks, fans, motherboard circuit breakers, and CPU module DC-to-DC converters. CODE EXAMPLE 7-4, an excerpt of output from the showenvironment command, indicates that the front panel Service Required LED is ON. When reviewing the complete output from the showenvironment command, check the state of all Service Required LEDs and verify that all components show a status of OK. See CODE EXAMPLE 4-1 for a sample of complete output from the showenvironment command.

CODE EXAMPLE 7-4 showenvironment Command Output

```
System Indicator Status:
---------------------------------------------------
SYS_FRONT.LOCATE     SYS_FRONT.SERVICE    SYS_FRONT.ACT
--------------------------------------------------------
OFF                  ON                   ON
.
.
.
sc>
```

4. **Examine the output of the** prtdiag -v **command. Type:**

```
sc> console
Enter #. to return to ALOM.
# /usr/platform/`uname -i`/sbin/prtdiag -v
```

The prtdiag -v command provides access to information stored by POST and OpenBoot Diagnostics tests. Any information from this command about the current state of the system is lost if the system is reset. When examining the output to identify problems, verify that all installed CPU modules, PCI cards, and memory modules are listed; check for any Service Required LEDs that are ON; and verify that the system PROM firmware is the latest version. CODE EXAMPLE 7-5 shows an excerpt of output from the `prtdiag -v` command. See CODE EXAMPLE 2-8 through CODE EXAMPLE 2-13 for the complete `prtdiag -v` output from a “healthy” Netra 440 server.

CODE EXAMPLE 7-5 `prtdiag -v` Command Output

```
System Configuration: Sun Microsystems  sun4u Netra 440
System clock frequency: 177 MHZ
Memory size: 4GB

==================================== CPUs ====================================
                      E$          CPU     CPU         Temperature         Fan
       CPU  Freq      Size        Impl.   Mask     Die     Ambient   Speed  Unit
       ---  --------  ----------  ------  ----  --------  --------  -----  ----
        0  1062 MHz  1MB         US-IIIi   2.3       -     -
        1  1062 MHz  1MB         US-IIIi   2.3       -     -

================================= IO Devices =================================
     Bus   Freq
Brd  Type  MHz   Slot        Name                          Model
---  ----  ----  ----------  ----------------------------  --------------------
 0   pci    66           MB  pci108e,abba (network)        SUNW,pci-ce
 0   pci    33           MB  isa/su (serial)
 0   pci    33           MB  isa/su (serial)
.
.
.

Memory Module Groups:
--------------------------------------------------
ControllerID   GroupID  Labels
--------------------------------------------------
0              0        C0/P0/B0/D0,C0/P0/B0/D1
0              1        C0/P0/B1/D0,C0/P0/B1/D1

Memory Module Groups:
--------------------------------------------------
ControllerID   GroupID  Labels
--------------------------------------------------
1              0        C1/P0/B0/D0,C1/P0/B0/D1
1              1        C1/P0/B1/D0,C1/P0/B1/D1
.
.
.
System PROM revisions:
----------------------
OBP 4.10.3 2003/05/02 20:25 Netra 440
OBDIAG 4.10.3 2003/05/02 20:26
#
```

5. **Check the system LEDs.**

6. **Check the** `/var/adm/messages` **file.**

   The following are clear indications of a failing part:

   - Warning messages from Solaris software about any hardware or software components
   - ALOM environmental messages about a failing part, including a fan or power supply

   If there is no clear indication of a failing part, investigate the installed applications, the network, or the disk configuration.

   If you have clear indications that a part has failed or is failing, replace that part as soon as possible.

   If the problem is a confirmed environmental failure, replace the fan or power supply as soon as possible.

   A system with a redundant configuration might still operate in a degraded state, but the stability and performance of the system will be affected. Since the system is still operational, attempt to isolate the fault using several methods and tools to ensure that the part you suspect as faulty really is causing the problems you are experiencing. See "Isolating Faults in the System" on page 32.

   For information about installing and replacing field-replaceable parts, refer to the *Netra 440 Server Service Manual* (817-3883-xx).

Troubleshooting a System After an Unexpected Reboot

This procedure assumes that the system console is in its default configuration, so that you are able to switch between the system controller and the system console. Refer to the *Netra 440 Server System Administration Guide*.

▼ To Troubleshoot a System After an Unexpected Reboot

1. **Log in to the system controller and access the** `sc>` **prompt.**

   For information, refer to the *Netra 440 Server System Administration Guide*.

2. **Examine the ALOM event log. Type:**

```
sc> showlogs
```

The ALOM event log shows system events such as reset events and LED indicator state changes that have occurred since the last system boot. CODE EXAMPLE 7-6 shows a sample event log, which indicates that the front panel Service Required LED is ON.

CODE EXAMPLE 7-6 `showlogs` Command Output

```
MAY 09 16:54:27 Sun-SFV440-a: 00060003: "SC System booted."
MAY 09 16:54:27 Sun-SFV440-a: 00040029: "Host system has shut down."
MAY 09 16:56:35 Sun-SFV440-a: 00060000: "SC Login: User admin Logged on."
MAY 09 16:56:54 Sun-SFV440-a: 00060000: "SC Login: User admin Logged on."
MAY 09 16:58:11 Sun-SFV440-a: 00040001: "SC Request to Power On Host."
MAY 09 16:58:11 Sun-SFV440-a: 00040002: "Host System has Reset"
MAY 09 16:58:13 Sun-SFV440-a: 0004000b: "Host System has read and cleared
                                         bootmode."
MAY 09 16:58:13 Sun-SFV440-a: 0004004f: "Indicator PS0.POK is now ON"
MAY 09 16:58:13 Sun-SFV440-a: 0004004f: "Indicator PS1.POK is now ON"
MAY 09 16:59:19 Sun-SFV440-a: 00040002: "Host System has Reset"
MAY 09 17:00:46 Sun-SFV440-a: 00040002: "Host System has Reset"
MAY 09 17:01:51 Sun-SFV440-a: 0004004f: "Indicator SYS_FRONT.SERVICE is now ON"
MAY 09 17:03:22 Sun-SFV440-a: 00040002: "Host System has Reset"
MAY 09 17:03:22 Sun-SFV440-a: 0004004f: "Indicator SYS_FRONT.SERVICE is now OFF"
MAY 09 17:03:24 Sun-SFV440-a: 0004000b: "Host System has read and cleared
                                         bootmode."
```

CODE EXAMPLE 7-6 `showlogs` Command Output *(Continued)*

```
MAY 09 17:04:30 Sun-SFV440-a: 00040002: "Host System has Reset"
MAY 09 17:05:59 Sun-SFV440-a: 00040002: "Host System has Reset"
MAY 09 17:06:40 Sun-SFV440-a: 0004004f: "Indicator SYS_FRONT.SERVICE is now ON"
MAY 09 17:07:44 Sun-SFV440-a: 0004004f: "Indicator SYS_FRONT.ACT is now ON"
sc>
```

Note – Time stamps for ALOM logs reflect UTC (Universal Time Coordinated) time, while time stamps for the Solaris OS reflect local (server) time. Therefore, a single event might generate messages that appear to be logged at different times in different logs.

---

3. **Examine the ALOM run log. Type:**

```
sc> consolehistory run -v
```

This command shows the log containing the most recent system console output of boot messages from the Solaris OS. When troubleshooting, examine the output for hardware or software errors logged by the operating environment on the system console. CODE EXAMPLE 7-7 shows sample output from the `consolehistory run -v` command.

CODE EXAMPLE 7-7 `consolehistory run -v` Command Output

```
May  9 14:48:22 Sun-SFV440-a rmclomv: SC Login: User admin Logged on.

#
# init 0
#
INIT: New run level: 0
The system is coming down.  Please wait.
System services are now being stopped.
Print services stopped.
May  9 14:49:18 Sun-SFV440-a last message repeated 1 time

May  9 14:49:38 Sun-SFV440-a syslogd: going down on signal 15

The system is down.
syncing file systems... done
Program terminated
{1} ok boot disk

Netra 440, No Keyboard
Copyright 1998-2003 Sun Microsystems, Inc.  All rights reserved.
OpenBoot 4.10.3, 4096 MB memory installed, Serial #53005571.
```

CODE EXAMPLE 7-7 `consolehistory run -v` Command Output *(Continued)*

```
Ethernet address 0:3:ba:28:cd:3, Host ID: 8328cd03.

Initializing     1MB of memory at addr        123fecc000 -

Initializing     1MB of memory at addr        123fe02000 -

Initializing    14MB of memory at addr        123f002000 -

Initializing    16MB of memory at addr        123e002000 -

Initializing   992MB of memory at addr        1200000000 -

Initializing  1024MB of memory at addr        1000000000 -

Initializing  1024MB of memory at addr         200000000 -

Initializing  1024MB of memory at addr                 0 -

Rebooting with command: boot disk
Boot device: /pci@1f,700000/scsi@2/disk@0,0  File and args:
\
SunOS Release 5.8 Version Generic_114696-04 64-bit
Copyright 1983-2003 Sun Microsystems, Inc.  All rights reserved.
Hardware watchdog enabled
Indicator SYS_FRONT.ACT is now ON
configuring IPv4 interfaces: ce0.
Hostname: Sun-SFV440-a
The system is coming up.  Please wait.
NIS domainname is Ecd.East.Sun.COM
Starting IPv4 router discovery.
starting rpc services: rpcbind keyserv ypbind done.
Setting netmask of lo0 to 255.0.0.0
Setting netmask of ce0 to 255.255.255.0
Setting default IPv4 interface for multicast: add net 224.0/4: gateway Sun-
SFV440-a
syslog service starting.
Print services started.
volume management starting.
The system is ready.

Sun-SFV440-a console login: May  9 14:52:57 Sun-SFV440-a rmclomv: NOTICE:
keyswitch change event - state = UNKNOWN

May  9 14:52:57 Sun-SFV440-a rmclomv: Keyswitch Position has changed to Unknown
state.
```

CODE EXAMPLE 7-7 `consolehistory run -v` Command Output *(Continued)*

```
May  9 14:52:58 Sun-SFV440-a rmclomv: NOTICE: keyswitch change event - state =
LOCKED

May  9 14:52:58 Sun-SFV440-a rmclomv: KeySwitch Position has changed to Locked
State.

May  9 14:53:00 Sun-SFV440-a rmclomv: NOTICE: keyswitch change event - state =
NORMAL

May  9 14:53:01 Sun-SFV440-a rmclomv: KeySwitch Position has changed to On State.

sc>
```

4. **Examine the ALOM boot log. Type:**

```
sc> consolehistory boot -v
```

The ALOM boot log contains boot messages from POST, OpenBoot firmware, and Solaris software from the server's most recent reset. When examining the output to identify a problem, check for error messages from POST and OpenBoot Diagnostics tests.

CODE EXAMPLE 7-8 shows the boot messages from POST. Note that POST returned no error messages. See "What POST Error Messages Tell You" on page 11 for a sample POST error message and more information about POST error messages.

CODE EXAMPLE 7-8 `consolehistory boot -v` Command Output (Boot Messages From POST)

```
Keyswitch set to diagnostic position.
@(#)OBP 4.10.3 2003/05/02 20:25 Netra 440
Clearing TLBs
Power-On Reset
Executing Power On SelfTest
0>@(#) Netra[TM] 440 POST 4.10.3 2003/05/04 22:08
       /export/work/staff/firmware_re/post/post-build-
4.10.3/Fiesta/system/integrated  (firmware_re)
0>Hard Powerup RST thru SW
0>CPUs present in system: 0 1
0>OBP->POST Call with %o0=00000000.01012000.
0>Diag level set to MIN.
0>MFG scrpt mode set NORM
0>I/O port set to TTYA.
0>Start selftest...
1>Print Mem Config
1>Caches : Icache is ON, Dcache is ON, Wcache is ON, Pcache is ON.
1>Memory interleave set to 0
```

CODE EXAMPLE 7-8 `consolehistory boot -v` Command Output (Boot Messages From POST) *(Continued)*

```
1>      Bank 0 1024MB : 00000010.00000000 -> 00000010.40000000.
1>      Bank 2 1024MB : 00000012.00000000 -> 00000012.40000000.
0>Print Mem Config
0>Caches : Icache is ON, Dcache is ON, Wcache is ON, Pcache is ON.
0>Memory interleave set to 0
0>      Bank 0 1024MB : 00000000.00000000 -> 00000000.40000000.
0>      Bank 2 1024MB : 00000002.00000000 -> 00000002.40000000.
0>INFO:
0>      POST Passed all devices.
0>POST: Return to OBP.
```

CODE EXAMPLE 7-9 shows the initialization of the OpenBoot PROM.

CODE EXAMPLE 7-9 `consolehistory boot -v` Command Output (OpenBoot PROM Initialization)

```
Keyswitch set to diagnostic position.
@(#)OBP 4.10.3 2003/05/02 20:25 Netra 440
Clearing TLBs
POST Results: Cpu 0000.0000.0000.0000
  %o0 = 0000.0000.0000.0000 %o1 = ffff.ffff.f00a.2b73 %o2 = ffff.ffff.ffff.ffff
POST Results: Cpu 0000.0000.0000.0001
  %o0 = 0000.0000.0000.0000 %o1 = ffff.ffff.f00a.2b73 %o2 = ffff.ffff.ffff.ffff
Membase: 0000.0000.0000.0000
MemSize: 0000.0000.0004.0000
Init CPU arrays Done
Probing /pci@1d,700000 Device 1  Nothing there
Probing /pci@1d,700000 Device 2  Nothing there
```

The following sample output shows the system banner.

CODE EXAMPLE 7-10 `consolehistory boot -v` Command Output (System Banner Display)

```
Netra 440, No Keyboard
Copyright 1998-2003 Sun Microsystems, Inc.  All rights reserved.
OpenBoot 4.10.3, 4096 MB memory installed, Serial #53005571.
Ethernet address 0:3:ba:28:cd:3, Host ID: 8328cd03.
```

The following sample output shows OpenBoot Diagnostics testing. See "What OpenBoot Diagnostics Error Messages Tell You" on page 20 for a sample OpenBoot Diagnostics error message and more information about OpenBoot Diagnostics error messages.

CODE EXAMPLE 7-11 `consolehistory boot -v` Command Output (OpenBoot Diagnostics Testing)

```
Running diagnostic script obdiag/normal

Testing /pci@1f,700000/network@1
Testing /pci@1e,600000/ide@d
Testing /pci@1e,600000/isa@7/flashprom@2,0
Testing /pci@1e,600000/isa@7/serial@0,2e8
Testing /pci@1e,600000/isa@7/serial@0,3f8
Testing /pci@1e,600000/isa@7/rtc@0,70
Testing /pci@1e,600000/isa@7/i2c@0,320:tests=
{gpio@0.42,gpio@0.44,gpio@0.46,gpio@0.48}
Testing /pci@1e,600000/isa@7/i2c@0,320:tests={hardware-monitor@0.5c}
Testing /pci@1e,600000/isa@7/i2c@0,320:tests={temperature-sensor@0.9c}
Testing /pci@1c,600000/network@2
Testing /pci@1f,700000/scsi@2,1
Testing /pci@1f,700000/scsi@2
```

The following sample output shows memory initialization by the OpenBoot PROM.

CODE EXAMPLE 7-12 `consolehistory boot -v` Command Output (Memory Initialization)

```
Initializing     1MB of memory at addr        123fe02000 -

Initializing    12MB of memory at addr        123f000000 -

Initializing  1008MB of memory at addr        1200000000 -

Initializing  1024MB of memory at addr        1000000000 -

Initializing  1024MB of memory at addr         200000000 -

Initializing  1024MB of memory at addr                 0 -

{1} ok boot disk
```

The following sample output shows the system booting and loading Solaris software

CODE EXAMPLE 7-13 `consolehistory boot -v` Command Output (System Booting and Loading Solaris Software)

```
Rebooting with command: boot disk
Boot device: /pci@1f,700000/scsi@2/disk@0,0  File and args:
Loading ufs-file-system package 1.4 04 Aug 1995 13:02:54.
FCode UFS Reader 1.11 97/07/10 16:19:15.
Loading: /platform/SUNW,Netra-440/ufsboot
Loading: /platform/sun4u/ufsboot
\
SunOS Release 5.8 Version Generic_114696-04 64-bit
Copyright 1983-2003 Sun Microsystems, Inc.  All rights reserved.
Hardware watchdog enabled
sc>
```

5. **Check the /`var/adm/messages` file for indications of an error.**

   Look for the following information about the system's state:

   - Any large gaps in the time stamp of Solaris software or application messages
   - Warning messages about any hardware or software components
   - Information from last root logins to determine whether any system administrators might be able to provide any information about the system state at the time of the hang

6. **If possible, check whether the system saved a core dump file.**

   Core dump files provide invaluable information to your support provider to aid in diagnosing any system problems. For further information about core dump files, see "The Core Dump Process" on page 103 and "Managing System Crash Information" in the *Solaris System Administration Guide*.

7. **Check the system LEDs.**

   You can use the ALOM system controller to check the state of the system LEDs. Refer to the *Netra 440 Server System Administration Guide* (817-3884-xx) for information about system LEDs.

8. **Examine the output of the `prtdiag -v` command. Type:**

```
sc> console
Enter #. to return to ALOM.
# /usr/platform/`uname -i`/sbin/prtdiag -v
```

The `prtdiag -v` command provides access to information stored by POST and OpenBoot Diagnostics tests. Any information from this command about the current state of the system is lost if the system is reset. When examining the output to identify problems, verify that all installed CPU modules, PCI cards, and memory modules are listed; check for any Service Required LEDs that are ON; and verify that the system PROM firmware is the latest version. CODE EXAMPLE 7-14 shows an excerpt of output from the `prtdiag -v` command. See CODE EXAMPLE 2-8 through CODE EXAMPLE 2-13 for the complete `prtdiag -v` output from a "healthy" Netra 440 server.

CODE EXAMPLE 7-14 `prtdiag -v` Command Output

```
System Configuration: Sun Microsystems  sun4u Netra 440
System clock frequency: 177 MHZ
Memory size: 4GB

==================================== CPUs ====================================
                      E$          CPU     CPU       Temperature         Fan
       CPU  Freq      Size        Impl.   Mask     Die    Ambient   Speed   Unit
       ---  --------  ----------  ------  ----  --------  --------  -----   ----
        0   1062 MHz  1MB         US-IIIi   2.3      -     -
        1   1062 MHz  1MB         US-IIIi   2.3      -     -

================================= IO Devices =================================
     Bus   Freq
Brd  Type  MHz   Slot        Name                          Model
---  ----  ----  ----------  ----------------------------  --------------------
 0   pci    66           MB  pci108e,abba (network)        SUNW,pci-ce
 0   pci    33           MB  isa/su (serial)
 0   pci    33           MB  isa/su (serial)

.
.
.
Memory Module Groups:
--------------------------------------------------
ControllerID   GroupID  Labels
--------------------------------------------------
0              0        C0/P0/B0/D0,C0/P0/B0/D1
0              1        C0/P0/B1/D0,C0/P0/B1/D1
.
.
.
System PROM revisions:
----------------------
OBP 4.10.3 2003/05/02 20:25 Netra 440
OBDIAG 4.10.3 2003/05/02 20:26
#
```

9. **Verify that all user and system processes are functional. Type:**

```
# ps -ef
```

Output from the `ps -ef` command shows each process, the start time, the run time, and the full process command-line options. To identify a system problem, examine the output for missing entries in the `CMD` column. CODE EXAMPLE 7-15 shows the `ps -ef` command output of a "healthy" Netra 440 server.

CODE EXAMPLE 7-15 `ps -ef` Command Output

```
     UID   PID  PPID  C    STIME TTY       TIME CMD
    root     0     0  0 14:51:32 ?         0:17 sched
    root     1     0  0 14:51:32 ?         0:00 /etc/init -
    root     2     0  0 14:51:32 ?         0:00 pageout
    root     3     0  0 14:51:32 ?         0:02 fsflush
    root   291     1  0 14:51:47 ?         0:00 /usr/lib/saf/sac -t 300
    root   205     1  0 14:51:44 ?         0:00 /usr/lib/lpsched
    root   312   148  0 14:54:33 ?         0:00 in.telnetd
    root   169     1  0 14:51:42 ?         0:00 /usr/lib/autofs/automountd
   user1   314   312  0 14:54:33 pts/1     0:00 -csh
    root    53     1  0 14:51:36 ?         0:00 /usr/lib/sysevent/syseventd
    root    59     1  0 14:51:37 ?         0:02 /usr/lib/picl/picld
    root   100     1  0 14:51:40 ?         0:00 /usr/sbin/in.rdisc -s
    root   131     1  0 14:51:40 ?      0:00 /usr/lib/netsvc/yp/ypbind -broadcast
    root   118     1  0 14:51:40 ?         0:00 /usr/sbin/rpcbind
    root   121     1  0 14:51:40 ?         0:00 /usr/sbin/keyserv
    root   148     1  0 14:51:42 ?         0:00 /usr/sbin/inetd -s
    root   218     1  0 14:51:44 ?         0:00 /usr/lib/power/powerd
    root   199     1  0 14:51:43 ?         0:00 /usr/sbin/nscd
    root   162     1  0 14:51:42 ?         0:00 /usr/lib/nfs/lockd
  daemon   166     1  0 14:51:42 ?         0:00 /usr/lib/nfs/statd
    root   181     1  0 14:51:43 ?         0:00 /usr/sbin/syslogd
    root   283     1  0 14:51:47 ?         0:00 /usr/lib/dmi/snmpXdmid -s Sun-
SFV440-a
    root   184     1  0 14:51:43 ?         0:00 /usr/sbin/cron
    root   235   233  0 14:51:44 ?         0:00 /usr/sadm/lib/smc/bin/smcboot
    root   233     1  0 14:51:44 ?         0:00 /usr/sadm/lib/smc/bin/smcboot
    root   245     1  0 14:51:45 ?         0:00 /usr/sbin/vold
    root   247     1  0 14:51:45 ?         0:00 /usr/lib/sendmail -bd -q15m
    root   256     1  0 14:51:45 ?        0:00 /usr/lib/efcode/sparcv9/efdaemon
    root   294   291  0 14:51:47 ?         0:00 /usr/lib/saf/ttymon
    root   304   274  0 14:51:51 ?         0:00 mibiisa -r -p 32826
    root   274     1  0 14:51:46 ?         0:00 /usr/lib/snmp/snmpdx -y -c
/etc/snmp/conf
    root   334   292  0 15:00:59 console   0:00 ps -ef
#
```

10. **Verify that all I/O devices and activities are still present and functioning. Type:**

```
# iostat -xtc
```

This command shows all I/O devices and reports activity for each device. To identify a problem, examine the output for installed devices that are not listed. CODE EXAMPLE 7-16 shows the `iostat -xtc` command output from a "healthy" Netra 440 server.

CODE EXAMPLE 7-16 `iostat -xtc` Command Output

```
                 extended device statistics                     tty         cpu
device      r/s   w/s  kr/s   kw/s wait actv svc_t  %w  %b tin tout us sy wt id
sd0         0.0   0.0    0.0   0.0  0.0  0.0   0.0   0  0    0 183   0  2  2 96
sd1          6.5    1.2   49.5    7.9  0.0  0.2   24.6   0   3
sd2          0.2    0.0    0.0    0.0  0.0  0.0    0.0   0   0
sd3          0.2    0.0    0.0    0.0  0.0  0.0    0.0   0   0
sd4          0.2    0.0    0.0    0.0  0.0  0.0    0.0   0   0
nfs1         0.0    0.0    0.0    0.0  0.0  0.0    0.0   0   0
nfs2         0.0    0.0    0.1    0.0  0.0  0.0    9.6   0   0
nfs3         0.1    0.0    0.6    0.0  0.0  0.0    1.4   0   0
nfs4         0.0    0.0    0.1    0.0  0.0  0.0    5.1   0   0
#
```

11. **Examine errors pertaining to I/O devices. Type:**

```
# iostat -E
```

This command reports on errors for each I/O device. To identify a problem, examine the output for any type of error that is more than 0. For example, in CODE EXAMPLE 7-17, `iostat -E` reports `Hard Errors: 2` for I/O device `sd0`.

CODE EXAMPLE 7-17 `iostat -E` Command Output

```
sd0      Soft Errors: 0 Hard Errors: 2 Transport Errors: 0
Vendor: TOSHIBA  Product: DVD-ROM SD-C2612 Revision: 1011 Serial No: 04/17/02
Size: 18446744073.71GB <-1 bytes>
Media Error: 0 Device Not Ready: 2 No Device: 0 Recoverable: 0
Illegal Request: 0 Predictive Failure Analysis: 0
sd1      Soft Errors: 0 Hard Errors: 0 Transport Errors: 0
Vendor: SEAGATE  Product: ST336607LSUN36G  Revision: 0207 Serial No:
3JA0BW6Y00002317
Size: 36.42GB <36418595328 bytes>
Media Error: 0 Device Not Ready: 0 No Device: 0 Recoverable: 0
Illegal Request: 0 Predictive Failure Analysis: 0
sd2      Soft Errors: 0 Hard Errors: 0 Transport Errors: 0
```

CODE EXAMPLE 7-17 `iostat -E` Command Output *(Continued)*

```
Vendor: SEAGATE  Product: ST336607LSUN36G  Revision: 0207 Serial No:
3JA0BRQJ00007316
Size: 36.42GB <36418595328 bytes>
Media Error: 0 Device Not Ready: 0 No Device: 0 Recoverable: 0
Illegal Request: 0 Predictive Failure Analysis: 0
sd3      Soft Errors: 0 Hard Errors: 0 Transport Errors: 0
Vendor: SEAGATE  Product: ST336607LSUN36G  Revision: 0207 Serial No:
3JA0BWL000002318
Size: 36.42GB <36418595328 bytes>
Media Error: 0 Device Not Ready: 0 No Device: 0 Recoverable: 0
Illegal Request: 0 Predictive Failure Analysis: 0
sd4      Soft Errors: 0 Hard Errors: 0 Transport Errors: 0
Vendor: SEAGATE  Product: ST336607LSUN36G  Revision: 0207 Serial No:
3JA0AGQS00002317
Size: 36.42GB <36418595328 bytes>
Media Error: 0 Device Not Ready: 0 No Device: 0 Recoverable: 0
Illegal Request: 0 Predictive Failure Analysis: 0
#
```

12. **Verify that any mirrored RAID devices are functioning. Type:**

```
# raidctl
```

This command shows the status of RAID devices. To identify a problem, examine the output for `Disk Status` that is not `OK`. For more information about configuring mirrored RAID devices, refer to “About Hardware Disk Mirroring” in the *Netra 440 Server System Administration Guide* (817-3884-xx).

CODE EXAMPLE 7-18 `raidctl` Command Output

```
# raidctl
RAID            RAID            RAID            Disk
Volume          Status          Disk            Status
------------------------------------------------------
c1t0d0          RESYNCING       c1t0d0          OK
                                c1t1d0          OK
#
```

13. **Run an exercising tool such as Sun VTS software or Hardware Diagnostic Suite.**

See Chapter 5 for information about exercising tools.

14. **If this is the first occurrence of an unexpected reboot and the system did not run POST as part of the reboot process, run POST.**

    If ASR is not enabled, now is a good time to enable ASR. ASR runs POST and OpenBoot Diagnostics tests automatically at reboot. With ASR enabled, you can save time diagnosing problems since POST and OpenBoot Diagnostics test results are already available after an unexpected reboot. Refer to the *Netra 440 Server System Administration Guide* (817-3884-xx) for more information about ASR and complete instructions for enabling ASR.

15. **Once troubleshooting is complete, schedule maintenance as necessary for any service actions.**

Troubleshooting Fatal Reset Errors and RED State Exceptions

This procedure assumes that the system console is in its default configuration, so that you are able to switch between the system controller and the system console. Refer to the *Netra 440 Server System Administration Guide*.

For more information about Fatal Reset errors and RED State Exceptions, see "Responding to Fatal Reset Errors and RED State Exceptions" on page 112. For a sample Fatal Reset error message, see CODE EXAMPLE 7-1. For a sample RED State Exception message, see CODE EXAMPLE 7-2.

1. **Log in to the system controller and access the `sc>` prompt.**

   For information, refer to the *Netra 440 Server System Administration Guide*.

2. **Examine the ALOM event log. Type:**

```
sc> showlogs
```

The ALOM event log shows system events such as reset events and LED indicator state changes that have occurred since the last system boot. CODE EXAMPLE 7-19 shows a sample event log, which indicates that the front panel Service Required LED is ON.

CODE EXAMPLE 7-19 `showlogs` Command Output

```
MAY 09 16:54:27 Sun-SFV440-a: 00060003: "SC System booted."
MAY 09 16:54:27 Sun-SFV440-a: 00040029: "Host system has shut down."
MAY 09 16:56:35 Sun-SFV440-a: 00060000: "SC Login: User admin Logged on."
MAY 09 16:56:54 Sun-SFV440-a: 00060000: "SC Login: User admin Logged on."
```

CODE EXAMPLE 7-19 `showlogs` Command Output *(Continued)*

```
MAY 09 16:58:11 Sun-SFV440-a: 00040001: "SC Request to Power On Host."
MAY 09 16:58:11 Sun-SFV440-a: 00040002: "Host System has Reset"
MAY 09 16:58:13 Sun-SFV440-a: 0004000b: "Host System has read and cleared
                                          bootmode."
MAY 09 16:58:13 Sun-SFV440-a: 0004004f: "Indicator PS0.POK is now ON"
MAY 09 16:58:13 Sun-SFV440-a: 0004004f: "Indicator PS1.POK is now ON"
MAY 09 16:59:19 Sun-SFV440-a: 00040002: "Host System has Reset"
MAY 09 17:00:46 Sun-SFV440-a: 00040002: "Host System has Reset"
MAY 09 17:01:51 Sun-SFV440-a: 0004004f: "Indicator SYS_FRONT.SERVICE is now ON"
MAY 09 17:03:22 Sun-SFV440-a: 00040002: "Host System has Reset"
MAY 09 17:03:22 Sun-SFV440-a: 0004004f: "Indicator SYS_FRONT.SERVICE is now OFF"
MAY 09 17:03:24 Sun-SFV440-a: 0004000b: "Host System has read and cleared
                                          bootmode."
MAY 09 17:04:30 Sun-SFV440-a: 00040002: "Host System has Reset"
MAY 09 17:05:59 Sun-SFV440-a: 00040002: "Host System has Reset"
MAY 09 17:06:40 Sun-SFV440-a: 0004004f: "Indicator SYS_FRONT.SERVICE is now ON"
MAY 09 17:07:44 Sun-SFV440-a: 0004004f: "Indicator SYS_FRONT.ACT is now ON"
sc>
```

Note – Time stamps for ALOM logs reflect UTC (Universal Time Coordinated) time, while time stamps for the Solaris OS reflect local (server) time. Therefore, a single event might generate messages that appear to be logged at different times in different logs.

3. **Examine the ALOM run log. Type:**

```
sc> consolehistory run -v
```

This command shows the log containing the most recent system console output of boot messages from the Solaris software. When troubleshooting, examine the output for hardware or software errors logged by the operating system on the system console. CODE EXAMPLE 7-20 shows sample output from the `consolehistory run -v` command.

CODE EXAMPLE 7-20 `consolehistory run -v` Command Output

```
May  9 14:48:22 Sun-SFV440-a rmclomv: SC Login: User admin Logged on.

#
# init 0
#
INIT: New run level: 0
The system is coming down.  Please wait.
System services are now being stopped.
```

CODE EXAMPLE 7-20 `consolehistory run -v` Command Output *(Continued)*

```
Print services stopped.
May  9 14:49:18 Sun-SFV440-a last message repeated 1 time

May  9 14:49:38 Sun-SFV440-a syslogd: going down on signal 15

The system is down.
syncing file systems... done
Program terminated
{1} ok boot disk

Netra 440, No Keyboard
Copyright 1998-2003 Sun Microsystems, Inc.  All rights reserved.
OpenBoot 4.10.3, 4096 MB memory installed, Serial #53005571.
Ethernet address 0:3:ba:28:cd:3, Host ID: 8328cd03.

Initializing     1MB of memory at addr        123fecc000 -

Initializing     1MB of memory at addr        123fe02000 -

Initializing    14MB of memory at addr        123f002000 -

Initializing    16MB of memory at addr        123e002000 -

Initializing   992MB of memory at addr        1200000000 -

Initializing  1024MB of memory at addr        1000000000 -

Initializing  1024MB of memory at addr         200000000 -

Initializing  1024MB of memory at addr                 0 -

Rebooting with command: boot disk
Boot device: /pci@1f,700000/scsi@2/disk@0,0  File and args:
\
SunOS Release 5.8 Version Generic_114696-04 64-bit
Copyright 1983-2003 Sun Microsystems, Inc.  All rights reserved.
Hardware watchdog enabled
Indicator SYS_FRONT.ACT is now ON
configuring IPv4 interfaces: ce0.
Hostname: Sun-SFV440-a
The system is coming up.  Please wait.
NIS domainname is Ecd.East.Sun.COM
Starting IPv4 router discovery.
starting rpc services: rpcbind keyserv ypbind done.
Setting netmask of lo0 to 255.0.0.0
Setting netmask of ce0 to 255.255.255.0
```

CODE EXAMPLE 7-20 `consolehistory run -v` Command Output *(Continued)*

```
Setting default IPv4 interface for multicast: add net 224.0/4: gateway Sun-
SFV440-a
syslog service starting.
Print services started.
volume management starting.
The system is ready.

Sun-SFV440-a console login: May  9 14:52:57 Sun-SFV440-a rmclomv: NOTICE:
keyswitch change event - state = UNKNOWN

May  9 14:52:57 Sun-SFV440-a rmclomv: Keyswitch Position has changed to Unknown
state.

May  9 14:52:58 Sun-SFV440-a rmclomv: NOTICE: keyswitch change event - state =
LOCKED

May  9 14:52:58 Sun-SFV440-a rmclomv: KeySwitch Position has changed to Locked
State.

May  9 14:53:00 Sun-SFV440-a rmclomv: NOTICE: keyswitch change event - state =
NORMAL

May  9 14:53:01 Sun-SFV440-a rmclomv: KeySwitch Position has changed to On State.

sc>
```

4. **Examine the ALOM boot log. Type:**

```
sc> consolehistory boot -v
```

The ALOM boot log contains boot messages from POST, OpenBoot firmware, and Solaris software from the server's most recent reset. When examining the output to identify a problem, check for error messages from POST and OpenBoot Diagnostics tests.

CODE EXAMPLE 7-21 shows the boot messages from POST. Note that POST returned no error messages. See "What POST Error Messages Tell You" on page 11 for a sample POST error message and more information about POST error messages.

CODE EXAMPLE 7-21 `consolehistory boot -v` Command Output (Boot Messages From POST)

```
Keyswitch set to diagnostic position.
@(#)OBP 4.10.3 2003/05/02 20:25 Netra 440
Clearing TLBs
Power-On Reset
Executing Power On SelfTest
```

CODE EXAMPLE 7-21 `consolehistory boot -v` Command Output (Boot Messages From POST)

```
0>@(#) Netra[TM] 440 POST 4.10.3 2003/05/04 22:08
       /export/work/staff/firmware_re/post/post-build-
4.10.3/Fiesta/system/integrated  (firmware_re)
0>Hard Powerup RST thru SW
0>CPUs present in system: 0 1
0>OBP->POST Call with %o0=00000000.01012000.
0>Diag level set to MIN.
0>MFG scrpt mode set NORM
0>I/O port set to TTYA.
0>
0>Start selftest...
1>Print Mem Config
1>Caches : Icache is ON, Dcache is ON, Wcache is ON, Pcache is ON.
1>Memory interleave set to 0
1>      Bank 0 1024MB : 00000010.00000000 -> 00000010.40000000.
1>      Bank 2 1024MB : 00000012.00000000 -> 00000012.40000000.
0>Print Mem Config
0>Caches : Icache is ON, Dcache is ON, Wcache is ON, Pcache is ON.
0>Memory interleave set to 0
0>      Bank 0 1024MB : 00000000.00000000 -> 00000000.40000000.
0>      Bank 2 1024MB : 00000002.00000000 -> 00000002.40000000.
0>INFO:
0>      POST Passed all devices.
0>
0>POST: Return to OBP.
```

The following output shows the initialization of the OpenBoot PROM.

CODE EXAMPLE 7-22 `consolehistory boot -v` Command Output (OpenBoot PROM Initialization)

```
Keyswitch set to diagnostic position.
@(#)OBP 4.10.3 2003/05/02 20:25 Netra 440
Clearing TLBs
POST Results: Cpu 0000.0000.0000.0000
  %o0 = 0000.0000.0000.0000 %o1 = ffff.ffff.f00a.2b73 %o2 = ffff.ffff.ffff.ffff
POST Results: Cpu 0000.0000.0000.0001
  %o0 = 0000.0000.0000.0000 %o1 = ffff.ffff.f00a.2b73 %o2 = ffff.ffff.ffff.ffff
Membase: 0000.0000.0000.0000
MemSize: 0000.0000.0004.0000
Init CPU arrays Done
Probing /pci@1d,700000 Device 1  Nothing there
Probing /pci@1d,700000 Device 2  Nothing there
```

The following sample output shows the system banner.

CODE EXAMPLE 7-23 `consolehistory boot -v` Command Output (System Banner Display)

```
Netra 440, No Keyboard
Copyright 1998-2003 Sun Microsystems, Inc.  All rights reserved.
OpenBoot 4.10.3, 4096 MB memory installed, Serial #53005571.
Ethernet address 0:3:ba:28:cd:3, Host ID: 8328cd03.
```

The following sample output shows OpenBoot Diagnostics testing. See "What OpenBoot Diagnostics Error Messages Tell You" on page 20 for a sample OpenBoot Diagnostics error message and more information about OpenBoot Diagnostics error messages.

CODE EXAMPLE 7-24 `consolehistory boot -v` Command Output (OpenBoot Diagnostics Testing)

```
Running diagnostic script obdiag/normal

Testing /pci@1f,700000/network@1
Testing /pci@1e,600000/ide@d
Testing /pci@1e,600000/isa@7/flashprom@2,0
Testing /pci@1e,600000/isa@7/serial@0,2e8
Testing /pci@1e,600000/isa@7/serial@0,3f8
Testing /pci@1e,600000/isa@7/rtc@0,70
Testing /pci@1e,600000/isa@7/i2c@0,320:tests=
{gpio@0.42,gpio@0.44,gpio@0.46,gpio@0.48}
Testing /pci@1e,600000/isa@7/i2c@0,320:tests={hardware-monitor@0.5c}
Testing /pci@1e,600000/isa@7/i2c@0,320:tests={temperature-sensor@0.9c}
Testing /pci@1c,600000/network@2
Testing /pci@1f,700000/scsi@2,1
Testing /pci@1f,700000/scsi@2
```

The following sample output shows memory initialization by the OpenBoot PROM.

CODE EXAMPLE 7-25 `consolehistory boot -v` Command Output (Memory Initialization)

```
Initializing     1MB of memory at addr        123fe02000 -

Initializing    12MB of memory at addr        123f000000 -

Initializing  1008MB of memory at addr        1200000000 -

Initializing  1024MB of memory at addr        1000000000 -

Initializing  1024MB of memory at addr         200000000 -

Initializing  1024MB of memory at addr                 0 -

{1} ok boot disk
```

The following sample output shows the system booting and loading the Solaris software.

CODE EXAMPLE 7-26 `consolehistory boot -v` Command Output (System Booting and Loading Solaris Software)

```
Rebooting with command: boot disk
Boot device: /pci@1f,700000/scsi@2/disk@0,0  File and args:
Loading ufs-file-system package 1.4 04 Aug 1995 13:02:54.
FCode UFS Reader 1.11 97/07/10 16:19:15.
Loading: /platform/SUNW,Netra-440/ufsboot
Loading: /platform/sun4u/ufsboot
\
SunOS Release 5.8 Version Generic_114696-04 64-bit
Copyright 1983-2003 Sun Microsystems, Inc.  All rights reserved.
Hardware watchdog enabled
sc>
```

5. **Check the /`var/adm/messages` file for indications of an error.**

   Look for the following information about the system's state:

   - Any large gaps in the time stamp of Solaris software or application messages
   - Warning messages about any hardware or software components
   - Information from last root logins to determine whether any system administrators might be able to provide any information about the system state at the time of the hang

6. **If possible, check whether the system saved a core dump file.**

   Core dump files provide invaluable information to your support provider to aid in diagnosing any system problems. For further information about core dump files, see "The Core Dump Process" on page 103 and "Managing System Crash Information" in the *Solaris System Administration Guide.*

7. **Check the system LEDs.**

   You can use the ALOM system controller to check the state of the system LEDs. Refer to the *Netra 440 Server System Administration Guide* (817-3884-xx) for information about system LEDs.

8. **Examine the output of the `prtdiag -v` command. Type:**

```
sc> console
Enter #. to return to ALOM.
# /usr/platform/`uname -i`/sbin/prtdiag -v
```

The `prtdiag -v` command provides access to information stored by POST and OpenBoot Diagnostics tests. Any information from this command about the current state of the system is lost if the system is reset. When examining the output to identify problems, verify that all installed CPU modules, PCI cards, and memory modules are listed; check for any Service Required LEDs that are ON; and verify that the system PROM firmware is the latest version. CODE EXAMPLE 7-27 shows an excerpt of output from the `prtdiag -v` command. See CODE EXAMPLE 2-8 through CODE EXAMPLE 2-13 for the complete `prtdiag -v` output from a “healthy” Netra 440 server.

CODE EXAMPLE 7-27 `prtdiag -v` Command Output

```
System Configuration: Sun Microsystems  sun4u Netra 440
System clock frequency: 177 MHZ
Memory size: 4GB

==================================== CPUs ====================================
                      E$          CPU     CPU       Temperature         Fan
       CPU  Freq      Size        Impl.   Mask     Die     Ambient   Speed   Unit
       ---  --------  ----------  ------  ----  --------  --------  -----  ----
         0  1062 MHz  1MB         US-IIIi   2.3      -      -
         1  1062 MHz  1MB         US-IIIi   2.3      -      -

================================= IO Devices =================================
     Bus   Freq
Brd  Type  MHz   Slot        Name                          Model
---  ----  ----  ----------  ----------------------------  --------------------
 0   pci    66           MB  pci108e,abba (network)        SUNW,pci-ce
 0   pci    33           MB  isa/su (serial)
 0   pci    33           MB  isa/su (serial)
.
.
.
Memory Module Groups:
--------------------------------------------------
ControllerID   GroupID  Labels
--------------------------------------------------
0              0        C0/P0/B0/D0,C0/P0/B0/D1
0              1        C0/P0/B1/D0,C0/P0/B1/D1
.
.
```

CODE EXAMPLE 7-27 `prtdiag -v` Command Output *(Continued)*

```
.
System PROM revisions:
----------------------
OBP 4.10.3 2003/05/02 20:25 Netra 440
OBDIAG 4.10.3 2003/05/02 20:26
#
```

9. **Verify that all user and system processes are functional. Type:**

```
# ps -ef
```

Output from the `ps -ef` command shows each process, the start time, the run time, and the full process command-line options. To identify a system problem, examine the output for missing entries in the `CMD` column. CODE EXAMPLE 7-28 shows the `ps -ef` command output of a "healthy" Netra 440 server.

CODE EXAMPLE 7-28 `ps -ef` Command Output

```
     UID   PID  PPID  C    STIME TTY      TIME CMD
    root     0     0  0 14:51:32 ?        0:17 sched
    root     1     0  0 14:51:32 ?        0:00 /etc/init -
    root     2     0  0 14:51:32 ?        0:00 pageout
    root     3     0  0 14:51:32 ?        0:02 fsflush
    root   291     1  0 14:51:47 ?        0:00 /usr/lib/saf/sac -t 300
    root   205     1  0 14:51:44 ?        0:00 /usr/lib/lpsched
    root   312   148  0 14:54:33 ?        0:00 in.telnetd
    root   169     1  0 14:51:42 ?        0:00 /usr/lib/autofs/automountd
   user1   314   312  0 14:54:33 pts/1    0:00 -csh
    root    53     1  0 14:51:36 ?        0:00 /usr/lib/sysevent/syseventd
    root    59     1  0 14:51:37 ?        0:02 /usr/lib/picl/picld
    root   100     1  0 14:51:40 ?        0:00 /usr/sbin/in.rdisc -s
   root   131     1  0 14:51:40 ?       0:00 /usr/lib/netsvc/yp/ypbind -broadcast
    root   118     1  0 14:51:40 ?        0:00 /usr/sbin/rpcbind
    root   121     1  0 14:51:40 ?        0:00 /usr/sbin/keyserv
    root   148     1  0 14:51:42 ?        0:00 /usr/sbin/inetd -s
    root   226     1  0 14:51:44 ?        0:00 /usr/lib/utmpd
    root   218     1  0 14:51:44 ?        0:00 /usr/lib/power/powerd
    root   199     1  0 14:51:43 ?        0:00 /usr/sbin/nscd
    root   162     1  0 14:51:42 ?        0:00 /usr/lib/nfs/lockd
  daemon   166     1  0 14:51:42 ?        0:00 /usr/lib/nfs/statd
    root   181     1  0 14:51:43 ?        0:00 /usr/sbin/syslogd
    root   283     1  0 14:51:47 ?        0:00 /usr/lib/dmi/snmpXdmid -s Sun-
SFV440-a
    root   184     1  0 14:51:43 ?        0:00 /usr/sbin/cron
    root   235   233  0 14:51:44 ?        0:00 /usr/sadm/lib/smc/bin/smcboot
    root   233     1  0 14:51:44 ?        0:00 /usr/sadm/lib/smc/bin/smcboot
```

CODE EXAMPLE 7-28 `ps -ef` Command Output *(Continued)*

```
    root   245     1  0 14:51:45 ?        0:00 /usr/sbin/vold
    root   247     1  0 14:51:45 ?        0:00 /usr/lib/sendmail -bd -q15m
    root   256     1  0 14:51:45 ?        0:00 /usr/lib/efcode/sparcv9/efdaemon
    root   294   291  0 14:51:47 ?        0:00 /usr/lib/saf/ttymon
    root   304   274  0 14:51:51 ?        0:00 mibiisa -r -p 32826
    root   274     1  0 14:51:46 ?        0:00 /usr/lib/snmp/snmpdx -y -c
/etc/snmp/conf
    root   334   292  0 15:00:59 console  0:00 ps -ef
    root   281     1  0 14:51:47 ?        0:00 /usr/lib/dmi/dmispd
    root   282     1  0 14:51:47 ?        0:00 /usr/dt/bin/dtlogin -daemon
    root   292     1  0 14:51:47 console  0:00 -sh
    root   324   314  0 14:54:51 pts/1    0:00 -sh
#
```

10. **Verify that all I/O devices and activities are still present and functioning. Type:**

```
# iostat -xtc
```

This command shows all I/O devices and reports activity for each device. To identify a problem, examine the output for installed devices that are not listed. CODE EXAMPLE 7-29 shows the `iostat -xtc` command output from a "healthy" Netra 440 server.

CODE EXAMPLE 7-29 `iostat -xtc` Command Output

```
                 extended device statistics                     tty        cpu
device      r/s    w/s   kr/s   kw/s wait actv  svc_t  %w  %b  tin tout  us sy wt id
sd0         0.0    0.0    0.0    0.0  0.0  0.0    0.0   0   0    0  183   0  2  2 96
sd1           6.5     1.2    49.5     7.9  0.0  0.2    24.6   0   3
sd2           0.2     0.0     0.0     0.0  0.0  0.0     0.0   0   0
sd3           0.2     0.0     0.0     0.0  0.0  0.0     0.0   0   0
sd4           0.2     0.0     0.0     0.0  0.0  0.0     0.0   0   0
nfs1          0.0     0.0     0.0     0.0  0.0  0.0     0.0   0   0
nfs2          0.0     0.0     0.1     0.0  0.0  0.0     9.6   0   0
nfs3          0.1     0.0     0.6     0.0  0.0  0.0     1.4   0   0
nfs4          0.0     0.0     0.1     0.0  0.0  0.0     5.1   0   0
#
```

11. **Examine errors pertaining to I/O devices. Type:**

```
# iostat -E
```

This command reports on errors for each I/O device. To identify a problem, examine the output for any type of error that is more than 0. For example, in CODE EXAMPLE 7-30, `iostat -E` reports `Hard Errors: 2` for I/O device `sd0`.

CODE EXAMPLE 7-30 `iostat -E` Command Output

```
sd0      Soft Errors: 0 Hard Errors: 2 Transport Errors: 0
Vendor: TOSHIBA  Product: DVD-ROM SD-C2612 Revision: 1011 Serial No: 04/17/02
Size: 18446744073.71GB <-1 bytes>
Media Error: 0 Device Not Ready: 2 No Device: 0 Recoverable: 0
Illegal Request: 0 Predictive Failure Analysis: 0
sd1      Soft Errors: 0 Hard Errors: 0 Transport Errors: 0
Vendor: SEAGATE  Product: ST336607LSUN36G  Revision: 0207 Serial No:
3JA0BW6Y00002317
Size: 36.42GB <36418595328 bytes>
Media Error: 0 Device Not Ready: 0 No Device: 0 Recoverable: 0
Illegal Request: 0 Predictive Failure Analysis: 0
sd2      Soft Errors: 0 Hard Errors: 0 Transport Errors: 0
Vendor: SEAGATE  Product: ST336607LSUN36G  Revision: 0207 Serial No:
3JA0BRQJ00007316
Size: 36.42GB <36418595328 bytes>
Media Error: 0 Device Not Ready: 0 No Device: 0 Recoverable: 0
Illegal Request: 0 Predictive Failure Analysis: 0
sd3      Soft Errors: 0 Hard Errors: 0 Transport Errors: 0
Vendor: SEAGATE  Product: ST336607LSUN36G  Revision: 0207 Serial No:
3JA0BWL000002318
Size: 36.42GB <36418595328 bytes>
Media Error: 0 Device Not Ready: 0 No Device: 0 Recoverable: 0
Illegal Request: 0 Predictive Failure Analysis: 0
sd4      Soft Errors: 0 Hard Errors: 0 Transport Errors: 0
Vendor: SEAGATE  Product: ST336607LSUN36G  Revision: 0207 Serial No:
3JA0AGQS00002317
Size: 36.42GB <36418595328 bytes>
Media Error: 0 Device Not Ready: 0 No Device: 0 Recoverable: 0
Illegal Request: 0 Predictive Failure Analysis: 0
#
```

12. **Check your system Product Notes and the SunSolve Online Web site for the latest information, driver updates, and Free Info Docs for the system.**

13. **Check the system's recent service history.**

    A system that has had several recent Fatal Reset errors and subsequent FRU replacements should be monitored closely to determine whether the recently replaced parts were, in fact, not faulty, and whether the actual faulty hardware has gone undetected.

Troubleshooting a System That Does Not Boot

A system might be unable to boot due to hardware or software problems. If you suspect that the system is unable to boot for software reasons, refer to "Troubleshooting Miscellaneous Software Problems" in the *Solaris System Administration Guide: Advanced Administration*. If you suspect the system is unable to boot due to a hardware problem, use the following procedure to determine the possible causes.

This procedure assumes that the system console is in its default configuration, so that you are able to switch between the system controller and the system console. Refer to the *Netra 440 Server System Administration Guide*.

1. **Log in to the system controller and access the `sc>` prompt.**

   For information, refer to the *Netra 440 Server System Administration Guide*.

2. **Examine the ALOM event log. Type:**

```
sc> showlogs
```

The ALOM event log shows system events such as reset events and LED indicator state changes that have occurred since the last system boot. To identify problems, examine the output for Service Required LEDs that are ON. CODE EXAMPLE 7-31 shows a sample event log, which indicates that the front panel Service Required LED is ON.

CODE EXAMPLE 7-31 `showlogs` Command Output

```
MAY 09 16:54:27 Sun-SFV440-a: 00060003: "SC System booted."
MAY 09 16:54:27 Sun-SFV440-a: 00040029: "Host system has shut down."
MAY 09 16:56:35 Sun-SFV440-a: 00060000: "SC Login: User admin Logged on."
MAY 09 16:56:54 Sun-SFV440-a: 00060000: "SC Login: User admin Logged on."
MAY 09 16:58:11 Sun-SFV440-a: 00040001: "SC Request to Power On Host."
MAY 09 16:58:11 Sun-SFV440-a: 00040002: "Host System has Reset"
MAY 09 16:58:13 Sun-SFV440-a: 0004000b: "Host System has read and cleared
                                          bootmode."
MAY 09 16:58:13 Sun-SFV440-a: 0004004f: "Indicator PS0.POK is now ON"
MAY 09 16:58:13 Sun-SFV440-a: 0004004f: "Indicator PS1.POK is now ON"
MAY 09 16:59:19 Sun-SFV440-a: 00040002: "Host System has Reset"
MAY 09 17:00:46 Sun-SFV440-a: 00040002: "Host System has Reset"
MAY 09 17:01:51 Sun-SFV440-a: 0004004f: "Indicator SYS_FRONT.SERVICE is now ON"
MAY 09 17:03:22 Sun-SFV440-a: 00040002: "Host System has Reset"
MAY 09 17:03:22 Sun-SFV440-a: 0004004f: "Indicator SYS_FRONT.SERVICE is now OFF"
MAY 09 17:03:24 Sun-SFV440-a: 0004000b: "Host System has read and cleared
                                          bootmode."
MAY 09 17:04:30 Sun-SFV440-a: 00040002: "Host System has Reset"
MAY 09 17:05:59 Sun-SFV440-a: 00040002: "Host System has Reset"
MAY 09 17:06:40 Sun-SFV440-a: 0004004f: "Indicator SYS_FRONT.SERVICE is now ON"
MAY 09 17:07:44 Sun-SFV440-a: 0004004f: "Indicator SYS_FRONT.ACT is now ON"
sc>
```

3. **Examine the ALOM run log. Type:**

```
sc> consolehistory run -v
```

This command shows the log containing the most recent system console output of boot messages from the Solaris OS. When troubleshooting, examine the output for hardware or software errors logged by the operating system on the system console. CODE EXAMPLE 7-32 shows sample output from the `consolehistory run -v` command.

CODE EXAMPLE 7-32 `consolehistory run -v` Command Output

```
May  9 14:48:22 Sun-SFV440-a rmclomv: SC Login: User admin Logged on.

#
# init 0
#
INIT: New run level: 0
The system is coming down.  Please wait.
System services are now being stopped.
Print services stopped.
May  9 14:49:18 Sun-SFV440-a last message repeated 1 time

```

CODE EXAMPLE 7-32 `consolehistory run -v` Command Output *(Continued)*

```
May  9 14:49:38 Sun-SFV440-a syslogd: going down on signal 15

The system is down.
syncing file systems... done
Program terminated
{1} ok boot disk

Netra 440, No Keyboard
Copyright 1998-2003 Sun Microsystems, Inc.  All rights reserved.
OpenBoot 4.10.3, 4096 MB memory installed, Serial #53005571.
Ethernet address 0:3:ba:28:cd:3, Host ID: 8328cd03.

Initializing     1MB of memory at addr        123fecc000 -

Initializing     1MB of memory at addr        123fe02000 -

Initializing    14MB of memory at addr        123f002000 -

Initializing    16MB of memory at addr        123e002000 -

Initializing   992MB of memory at addr        1200000000 -

Initializing  1024MB of memory at addr        1000000000 -

Initializing  1024MB of memory at addr         200000000 -

Initializing  1024MB of memory at addr                 0 -

Rebooting with command: boot disk
Boot device: /pci@1f,700000/scsi@2/disk@0,0  File and args:
\
SunOS Release 5.8 Version Generic_114696-04 64-bit
Copyright 1983-2003 Sun Microsystems, Inc.  All rights reserved.
Hardware watchdog enabled
Indicator SYS_FRONT.ACT is now ON
configuring IPv4 interfaces: ce0.
Hostname: Sun-SFV440-a
The system is coming up.  Please wait.
NIS domainname is Ecd.East.Sun.COM
Starting IPv4 router discovery.
starting rpc services: rpcbind keyserv ypbind done.
Setting netmask of lo0 to 255.0.0.0
Setting netmask of ce0 to 255.255.255.0
Setting default IPv4 interface for multicast: add net 224.0/4: gateway Sun-
SFV440-a
syslog service starting.
Print services started.
```

CODE EXAMPLE 7-32 `consolehistory run -v` Command Output *(Continued)*

```
volume management starting.
The system is ready.

Sun-SFV440-a console login: May  9 14:52:57 Sun-SFV440-a rmclomv: NOTICE:
keyswitch change event - state = UNKNOWN

May  9 14:52:57 Sun-SFV440-a rmclomv: Keyswitch Position has changed to Unknown
state.

May  9 14:52:58 Sun-SFV440-a rmclomv: NOTICE: keyswitch change event - state =
LOCKED

May  9 14:52:58 Sun-SFV440-a rmclomv: KeySwitch Position has changed to Locked
State.

May  9 14:53:00 Sun-SFV440-a rmclomv: NOTICE: keyswitch change event - state =
NORMAL

May  9 14:53:01 Sun-SFV440-a rmclomv: KeySwitch Position has changed to On State.

sc>
```

Note – Time stamps for ALOM logs reflect UTC (Universal Time Coordinated) time, while time stamps for the Solaris OS reflect local (server) time. Therefore, a single event might generate messages that appear to be logged at different times in different logs.

Note – The ALOM system controller runs independently from the system and uses standby power from the server. Therefore, ALOM firmware and software continue to function when power to the machine is turned off.

4. **Examine the ALOM boot log. Type:**

```
sc> consolehistory boot -v
```

The ALOM boot log contains boot messages from POST, OpenBoot firmware, and the Solaris software from the server's most recent reset. When examining the output to identify a problem, check for error messages from POST and OpenBoot Diagnostics tests.

CODE EXAMPLE 7-33 shows the boot messages from POST. Note that POST returned no error messages. See "What POST Error Messages Tell You" on page 11 for a sample POST error message and more information about POST error messages.

CODE EXAMPLE 7-33 `consolehistory boot -v` Command Output (Boot Messages From POST)

```
Keyswitch set to diagnostic position.
@(#)OBP 4.10.3 2003/05/02 20:25 Netra 440
Clearing TLBs
Power-On Reset
Executing Power On SelfTest

0>@(#) Netra[TM] 440 POST 4.10.3 2003/05/04 22:08
       /export/work/staff/firmware_re/post/post-build-
4.10.3/Fiesta/system/integrated  (firmware_re)
0>Hard Powerup RST thru SW
0>CPUs present in system: 0 1
0>OBP->POST Call with %o0=00000000.01012000.
0>Diag level set to MIN.
0>MFG scrpt mode set NORM
0>I/O port set to TTYA.
0>
0>Start selftest...
1>Print Mem Config
1>Caches : Icache is ON, Dcache is ON, Wcache is ON, Pcache is ON.
1>Memory interleave set to 0
1>      Bank 0 1024MB : 00000010.00000000 -> 00000010.40000000.
1>      Bank 2 1024MB : 00000012.00000000 -> 00000012.40000000.
0>Print Mem Config
0>Caches : Icache is ON, Dcache is ON, Wcache is ON, Pcache is ON.
0>Memory interleave set to 0
0>      Bank 0 1024MB : 00000000.00000000 -> 00000000.40000000.
0>      Bank 2 1024MB : 00000002.00000000 -> 00000002.40000000.
0>INFO:
0>      POST Passed all devices.
0>
0>POST: Return to OBP.
```

5. **Turn the system control rotary switch to the Diagnostics position.**

6. **Power on the system.**

   If the system does not boot, the system might have a basic hardware problem. If you have not made any recent hardware changes to the system, contact your authorized service provider.

7. **If the system gets to the `ok` prompt but does not load the operating system, you might need to change the `boot-device` setting in the system firmware.**

   See "Using OpenBoot Information Commands" on page 83 for information about using the `probe` commands. You can use the `probe` commands to display information about active SCSI and IDE devices.

   For information on changing the default boot device, refer to the *Solaris System Administration Guide: Basic Administration*.

   a. **Try to load the operating system for a single user from a CD.**

      Place a valid Solaris OS CD into the system DVD–ROM or CD–ROM drive and enter `boot cdrom -s` from the `ok` prompt.

   b. **If the system boots from the CD and loads the operating system, check the following:**

      - If the system normally boots from a system hard disk, check the system disk for problems and a valid boot image.
      - If the system normally boots from the network, check the system network configuration, the system Ethernet cables, and the system network card.

   c. **If the system gets to the `ok` prompt but does not load the operating system from the CD, check the following:**

      - OpenBoot variable settings (`boot-device`, `diag-device`, and `auto-boot?`).
      - OpenBoot PROM device tree. See "`show-devs` Command" on page 23 for more information.
      - That the banner was displayed before the `ok` prompt.
      - Any diagnostic test failure or other hardware failure message before the `ok` prompt was displayed.

Troubleshooting a System That Is Hanging

This procedure assumes that the system console is in its default configuration, so that you are able to switch between the system controller and the system console. Refer to the *Netra 440 Server System Administration Guide*.

▼ To Troubleshoot a System That Is Hanging

1. **Verify that the system is hanging.**

   a. **Type the** `ping` **command to determine whether there is any network activity.**

   b. **Type the** `ps -ef` **command to determine whether any other user sessions are active or responding.**

      If another user session is active, use it to review the contents of the `/var/adm/messages` file for any indications of the system problem.

   c. **Try to access the system console through the ALOM system controller.**

      If you can establish a working system console connection, the problem might not be a true hang but might instead be a network-related problem. For suspected network problems, use the `ping`, `rlogin`, or `telnet` commands to reach another system that is on the same sub-network, hub, or router. If NFS services are served by the affected system, determine whether NFS activity is present on other systems.

   d. **Change the system control rotary switch position while observing the system console.**

      For example, turn the rotary switch from the Normal position to the Diagnostics position, or from the Locked position to the Normal position. If the system console logs the change of rotary switch position, the system is not fully hung.

2. **If there are no responding user sessions, record the state of the system LEDs.**

   The system LEDs might indicate a hardware failure in the system. You can use the ALOM system controller to check the state of the system LEDs. Refer to the *Netra 440 Server System Administration Guide* (817-3884-xx) for more information about system LEDs.

3. **Attempt to bring the system to the `ok` prompt.**

   For instructions, refer to the *Netra 440 Server System Administration Guide*.

   If the system can get to the `ok` prompt, then the system hang can be classified as a soft hang. Otherwise, the system hang can be classified as a hard hang. See "Responding to System Hang States" on page 111 for more information.

4. **If the preceding step failed to bring the system to the `ok` prompt, execute an externally initiated reset (XIR).**

   Executing an XIR resets the system and preserves the state of the system before it resets, so that indications and messages about transient errors might be saved.

   An XIR is the equivalent of issuing a direct hardware reset. For further information about XIR, refer to the *Netra 440 Server System Administration Guide*.

5. **If an XIR brings the system to the `ok` prompt, do the following.**

   a. **Issue the `printenv` command.**

      This command displays the settings of the OpenBoot configuration variables.

   b. **Set the `auto-boot?` variable to `true`, the `diag-switch?` variable to `true`, the `diag-level` variable to `max`, and the `post-trigger` and `obdiag-trigger` variables to `all-resets`.**

   c. **Issue the `sync` command to obtain a core dump file.**

      Core dump files provide invaluable information to your support provider to aid in diagnosing any system problems. For further information about core dump files, see "The Core Dump Process" on page 103 and "Managing System Crash Information" in the *Solaris System Administration Guide*, which is part of the Solaris System Administrator Collection.

      The system reboots automatically provided that the OpenBoot configuration `auto-boot?` variable is set to `true` (default value).

Note – Steps 3, 4, and 5 occur automatically when the hardware watchdog mechanism is enabled.

---

6. **If an XIR failed to bring the system to the `ok` prompt, follow these steps:**

   a. **Turn the system control rotary switch to the Diagnostics position.**

      This forces the system to run POST and OpenBoot Diagnostics tests during system startup.

   b. **Press the system Power button for five seconds.**

      This causes an immediate hardware shutdown.

   c. **Wait at least 30 seconds; then power on the system by pressing the Power button.**

Note – You can also use the ALOM system controller to set the POST and OpenBoot Diagnostics levels, and to power off and reboot the system. Refer to the *Advanced Lights Out Manager Software User's Guide for the Netra 440 Server* (817-5481-xx).

7. **Use the POST and OpenBoot Diagnostics tests to diagnose system problems.**

   When the system initiates the startup sequence, it will run POST and OpenBoot Diagnostics tests. See "Isolating Faults Using POST Diagnostics" on page 60 and "Isolating Faults Using Interactive OpenBoot Diagnostics Tests" on page 62.

8. **Review the contents of the** `/var/adm/messages` **file.**

   Look for the following information about the system's state:

   - Any large gaps in the time stamp of Solaris software or application messages
   - Warning messages about any hardware or software components
   - Information from last root logins to determine whether any system administrators might be able to provide any information about the system state at the time of the hang

9. **If possible, check whether the system saved a core dump file.**

   Core dump files provide invaluable information to your support provider to aid in diagnosing any system problems. For further information about core dump files, see "The Core Dump Process" on page 103 and "Managing System Crash Information" in the *Solaris System Administration Guide*, which is part of the Solaris System Administrator Collection.

Index