Sun Fire™ V890 Server Diagnostics Guide

Sun Microsystems, Inc.
www.sun.com

Part No. 817-5950-11
October 2004, Revision 01

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

Please
Recycle

Contents

Preface

The *Sun Fire V890 Server Diagnostics Guide* is intended for experienced system administrators. It includes descriptive information about the Sun Fire™ V890 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 and network concepts and terms, and have advanced familiarity with the Solaris™ Operating System.

To use the information in this document fully, you must have thorough knowledge of the topics discussed in this book:

- *Sun Fire V890 Server Owner's Guide*

Using UNIX Commands

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

- Software documentation that you received with your system
- Solaris Operating System 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
Installation	*Sun Fire V890 Server Rackmounting Guide* *Installation Instructions for Solaris* *Solaris (SPARC Platform Edition) Installation Guide* *Solaris (SPARC Platform Edition) Installation Release Notes* *Solaris Sun Hardware Platform Guide* *Solaris Installation Guide* *Solaris Advanced Installation Guide*
Owner's Guide	*Sun Fire V890 Server Owner's Guide*
Service	*Sun Fire V890 Server Service Manual*
Late-Breaking Information	*Sun Fire V890 Server Product Notes* *Solaris Release Notes* *Solaris Release Notes Supplement for Sun Hardware*
System Diagnostics	*SunVTS User's Guide* *SunVTS Test Reference Manual* *SunVTS Quick Reference Card* *SunVTS Documentation Supplement*
System Management	*Sun Management Center Software Installation Guide* *Sun Management Center Software User's Guide* *Sun Management Center Software Release Notes* *Sun Management Center Supplement for Workgroup Servers*
System Administration	*Solaris System Administrator Documentation* *Platform Notes: The* `eri` *FastEthernet Device Driver* *Platform Notes: The Sun Gigabit Ethernet Device Driver* *Platform Notes: Using* `luxadm` *Software* *Sun Fire V890 Server Dynamic Reconfiguration User's Guide* *OpenBoot 4.x Command Reference Manual* *OpenBoot 4.x Quick Reference* *OpenBoot PROM Enhancements for Diagnostic Operation*
Remote System Monitoring and Control	*Sun Remote System Control (RSC) 2.2.2 User's Guide*

Accessing Sun Documentation

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

http://www.sun.com/documentation

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:

Sun Fire V890 Server Diagnostics Guide, part number 817-5950-11

CHAPTER 1

Diagnostic Tools

This chapter covers the diagnostic tools that are available for the Sun Fire V890 server and provides instructions on how to use these tools.

The following tasks are covered in this chapter:

The following information is also included:

Note – The procedures in this chapter assume that you are familiar with the OpenBoot™ firmware and that you know how to enter the OpenBoot environment. For more information about the OpenBoot firmware, see the *OpenBoot 4.x Command Reference Manual*. An online version of the manual is included with the *OpenBoot Collection AnswerBook* that ships with Solaris software.

About Diagnostic Tools

The system provides both firmware-based and software-based diagnostic tools to help you identify and isolate hardware problems. These tools include:

- Power-on self test (POST) diagnostics
- OpenBoot Diagnostics
- Sun Validation Test Suite (SunVTS™) software
- Sun™ Management Center software
- Sun Remote System Control (RSC) software

POST diagnostics verify the core functionality of the system, including the motherboard, CPU/Memory board, DIMMs, and PCI slots. You can run POST even if the system is unable to boot. For more information about POST, see "About POST Diagnostics" on page 4 and "Running POST Diagnostics" on page 8.

OpenBoot Diagnostics tests focus on system I/O and peripheral devices. Like POST, you can run OpenBoot Diagnostics even if the system is unable to boot. For more information about OpenBoot Diagnostics, see "About OpenBoot Diagnostics" on page 17 and "Running OpenBoot Diagnostics" on page 31.

SunVTS system exerciser is a graphics-oriented UNIX application that handles the continuous exercising of system resources and internal and external peripheral equipment. For more information about SunVTS software, see "About SunVTS Software" on page 38.

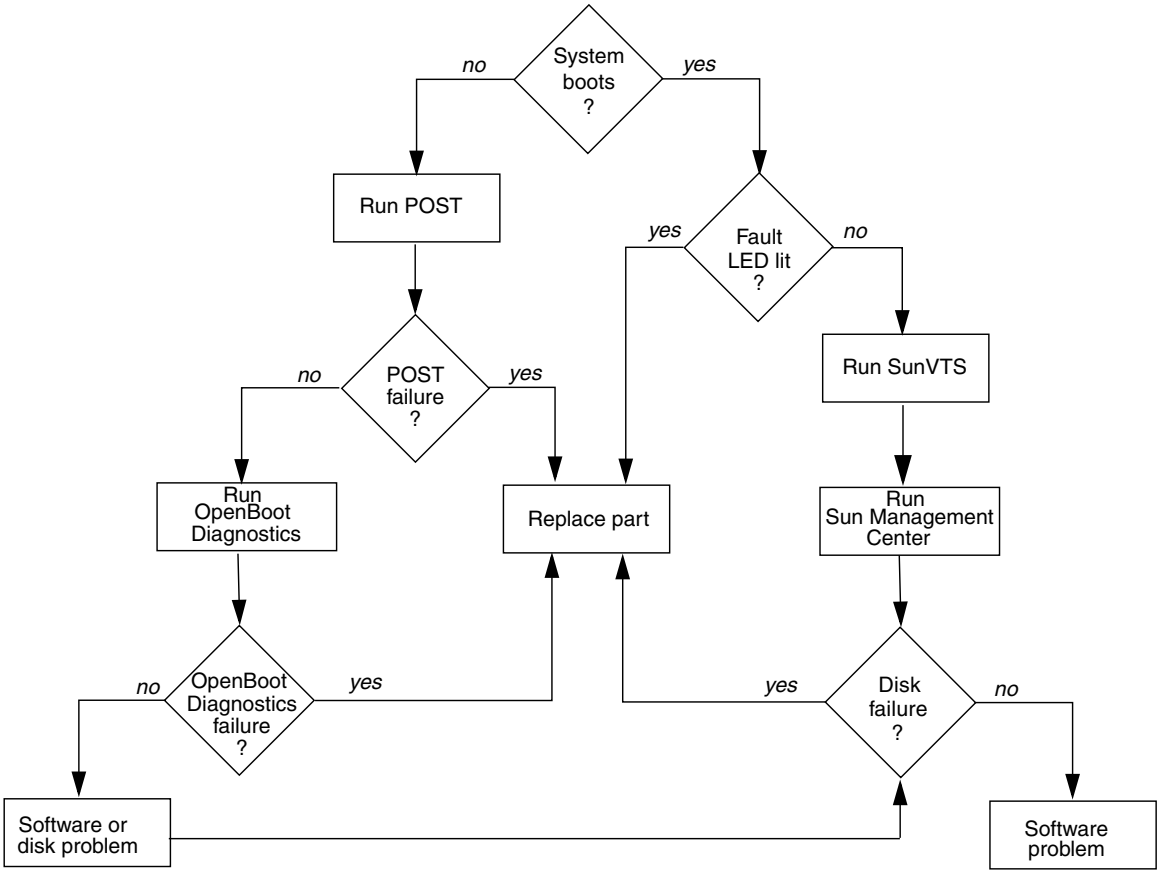
Sun Management Center (formerly Sun Enterprise SyMON™) software enables you to monitor the system hardware status and operating system performance of your server. For more information about Sun Management Center software, see "About Sun Management Center Software" on page 45.

Remote System Control (RSC) software is a server management tool that provides remote system administration for geographically distributed or physically inaccessible systems. For more information about RSC, see "About Sun Remote System Control Software" on page 46.

Which method or tool you use to diagnose system problems depends on the nature of those problems:

- If your system is unable to boot its operating system software, you need to run POST and OpenBoot Diagnostics tests.
- If your system is "healthy" enough to start up and load its operating system software, you can use Sun Management Center software and SunVTS software to diagnose system problems.
- If your system is at a remote location, use RSC software to diagnose problems remotely.

The following chart shows which tools you can use to diagnose hardware and software problems.



About POST Diagnostics

The POST diagnostic code resides in the OpenBoot PROM on the system I/O board. When you power on the system, POST runs automatically under certain conditions. For information about running POST, see "Running POST Diagnostics" on page 8.

POST tests the following system components:

- CPU/Memory board
- PCI slots (POST tests for shorts only)
- DIMMs
- Motherboard

OpenBoot diagnostic configuration variables, stored in the system nonvolatile random access memory (NVRAM), enable you to control certain aspects of POST testing. For information about the configuration variables, see "OpenBoot Configuration Variables for POST" on page 5.

POST reports its test results through detailed diagnostic and error messages. See "OpenBoot Configuration Variables for POST" on page 5 for information about diagnostic and error messages.

By default, POST displays diagnostic and error messages through a `tip` connection or a local ASCII terminal attached to the system's serial port A (`ttya`). You can also redirect POST output to display remotely on a Remote System Control (RSC) console. If you redirect POST output to an RSC console, POST results will not display locally. See "Displaying POST Diagnostic Results" on page 11 for information about redirecting POST output to an RSC console.

The system controller card runs its own POST diagnostics separately from the main POST diagnostics. RSC POST tests the basic functions of the system controller card. To view detailed diagnostic and error messages from RSC POST, you must attach an ASCII terminal directly to the RSC serial port before running RSC POST. For more information about RSC POST, see the *Sun Remote System Control (RSC) User's Guide*.

OpenBoot Configuration Variables for POST

The following table lists and describes the OpenBoot configuration variables that enable you to control the operation of POST.

Note – Both POST and OpenBoot Diagnostics use the settings of the configuration variables `diag-level`, `diag-switch?`, and `diag-trigger`. Changing the values of these variables will affect both POST and OpenBoot Diagnostics operation. See "OpenBoot Configuration Variables for OpenBoot Diagnostics" on page 18 for a complete listing and description of the configuration variables that control OpenBoot Diagnostics testing.

Variable	Setting	Description	Default
`diag-level`		Determines the level of testing executed.	`max`
	`off`	May perform initialization, but no testing.	
	`min`	Performs limited testing.	
	`max`	Runs extensive tests.	
	`menus`	Forces POST to enter interactive mode, providing access to advanced debugging features (for manufacturing use only).	
`diag-switch?`		Controls diagnostic execution in normal mode.	`false`
	`true`	Diagnostics are *only* executed on power-on reset events, but the level of test coverage, verbosity, and output is determined by user-defined settings.	
	`false`	Diagnostics are executed upon next system reset, but only for those class of reset events specified by the OpenBoot configuration variable `diag-trigger`. The level of test coverage, verbosity, and output is determined by user-defined settings.	

Variable	Setting	Description	Default
diag-trigger		Specifies the class of reset event that causes diagnostics to run automatically. **Note**: Both POST and OpenBoot Diagnostics run at the specified reset event if the variable diag-script is set to normal or all. If diag-script is set to none, only POST runs.	power-on-reset error-reset
	none	Diagnostic tests are not executed.	
	error-reset	Reset that is caused by certain hardware error events such as RED State Exception Reset, Watchdog Resets, Software-Instruction Reset, or Hardware Fatal Reset.	
	power-on-reset	Reset that is caused by power cycling the system.	
	user-reset	Reset that is initiated by an operating system panic or by userinitiated commands from OpenBoot (reset-all or boot) or from Solaris (reboot, shutdown, or init).	
	all-resets	Any kind of reset.	

To display the current and default values of all OpenBoot configuration variables, use the printenv command without specifying a variable.

The following is sample output from the `printenv` command.

```
ok printenv
Variable Name         Value                          Default Value

test-args
diag-passes           1                               1
local-mac-address?    true                            false
scsi-initiator-id     7                               7
oem-logo                                              No default
oem-logo?             false                           false
oem-banner                                            No default
oem-banner?           false                           false
ansi-terminal?        true                            true
screen-#columns       80                              80
screen-#rows          34                              34
ttyb-rts-dtr-off      false                           false
ttyb-ignore-cd        true                            true
ttya-rts-dtr-off      false                           false
ttya-ignore-cd        true                            true
ttyb-mode             9600,8,n,1,-                    9600,8,n,1,-
ttya-mode             9600,8,n,1,-                    9600,8,n,1,-
output-device         ttya                            screen
input-device          ttya                            keyboard
auto-boot-on-error?   true                            true
load-base             16384                           16384
auto-boot?            false                           true
boot-command          boot                            boot
diag-file
diag-device           disk net                        net
boot-file
boot-device           /pci@8,600000/SUNW,qlc@2 ...    disk net
use-nvramrc?          false                           false
nvramrc
security-mode         none                            No default
security-password                                     No default
security-#badlogins   0                               No default
verbosity             debug                           normal
fcode-debug?          false                           false
diag-out-console      false                           false
diag-trigger          none                            error-reset
power-on-res ...
service-mode?         false                           false
diag-script           none                            normal
diag-level            off                             max
diag-switch?          false                           false
error-reset-recovery  sync                            sync
```

To display the current and default values of a specific OpenBoot configuration variable, specify the `printenv` command and the variable name at the `ok` prompt.

```
ok printenv diag-switch?
diag-switch? =         true
ok
```

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

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

Running POST Diagnostics

When you power on the system, POST runs automatically under either of the following conditions:

- The front panel keyswitch is set to the Diagnostics position and the OpenBoot configuration variable `diag-level` is set to its default value (or to any valid setting other than `off`).
- The OpenBoot configuration variable `diag-switch?` is set to `true` and both `diag-level` and `diag-trigger` are set to their default values (or to any valid setting other than `off`).

Note – The default value for `diag-switch?` is `false`. Therefore, if all OpenBoot configuration variables are set to their default values, POST does not run unless the keyswitch is set to the Diagnostics position or `service-mode?` is set to `true`. For maximum test coverage, set `diag-level` variable to `max` prior to starting POST diagnostics.

You can also configure POST to run automatically after specific types of reset events by setting the values of the OpenBoot configuration variables `diag-switch?` and `diag-trigger`, as shown in the following table. Note that `diag-level` must be set to any valid value other than `none`. For more information, see "OpenBoot Configuration Variables for POST" on page 5.

Reset Event	POST Runs Automatically If...
Any power-on reset, including RSC-initiated power-on resets	The front panel keyswitch is set to the Diagnostics position OR `diag-switch?` is set to `true` and `diag-trigger` is set to any setting other than `none`
Any automatic reset triggered by a hardware error, including all operating system panics and watchdog reset events	`diag-switch?` is set to `true` and `diag-trigger` is set to `error-reset` or `soft-reset`
Any user-initiated reset event	`diag-switch?` is set to `true` and `diag-trigger` is set to `soft-reset`

How to Run POST Diagnostics

This procedure explains how to run POST diagnostics. There are two parts to this procedure:

- "Initiating POST Diagnostics" on page 10
- "Displaying POST Diagnostic Results" on page 11

Following this procedure are:

- "Sample POST Diagnostic Output" on page 12
- "Sample POST Error Messages" on page 14
- "Sample Summary of POST Results" on page 16

Before You Begin

You can view POST status and error messages on a local ASCII terminal or through a `tip` connection. You can also view messages remotely on an RSC console. To view POST diagnostic messages remotely on an RSC console, you need to configure the RSC software before starting POST. For information about using the RSC software, see the *Sun Remote System Control (RSC) User's Guide*. For information about setting up an alphanumeric terminal or establishing a `tip` connection, see the *Sun Fire V890 Server Owner's Guide*.

Note – By default, POST diagnostics output displays locally on an attached terminal or through a `tip` connection. However, if diagnostics output is redirected to an RSC console, the output will not display locally until it is directed back to the local terminal or `tip` connection. For information about directing POST output to an RSC console or to a local terminal or `tip` connection, see the *Sun Remote System Control (RSC) User's Guide* and "Displaying POST Diagnostic Results" on page 11.

Initiating POST Diagnostics

To start POST diagnostics, follow these steps:

1. **Turn the keyswitch to the Diagnostics position.**

   For information about the keyswitch position, see the *Sun Fire V890 Server Owner's Guide*.

2. **Press the Power button.**

   The system runs the POST diagnostics.

   POST displays status and error messages locally on an attached terminal, through a `tip` connection, or on an RSC console (if POST output has been redirected to the RSC console). For more information, see "Displaying POST Diagnostic Results" on page 11.

   Upon completion of POST, the system will run OpenBoot Diagnostics. For more information about OpenBoot Diagnostics, see "About OpenBoot Diagnostics" on page 17.

Displaying POST Diagnostic Results

As POST runs, it displays diagnostic status messages locally on an attached terminal, through a `tip` connection, or on an RSC console (if POST output has been redirected to the RSC console). By default, POST output displays locally on an attached terminal or through a `tip` connection.

To redirect POST output to an RSC console, follow these steps:

1. **Type the following commands at the `ok` prompt:**

```
ok diag-console rsc
ok setenv input-device rsc-console
ok setenv output-device rsc-console
```

2. **To cause the changes to take effect, power cycle the system, or type:**

```
ok reset-all
```

If you redirect POST output to an RSC console, the POST results will not display locally on an attached terminal or through a `tip` connection. To redirect POST output to the terminal or `tip` connection, issue the `diag-console` command as shown in the following example:

```
ok diag-console ttya
ok reset-all
```

See the *Sun Remote System Control (RSC) User's Guide* for more information.

When POST starts, it selects a master CPU to control test execution and error handling. If the master CPU fails, the CPU takes itself offline, and POST selects a new master if another CPU exists in the system.

The level of POST testing depends on the setting of the variable `diag-level`. See "OpenBoot Configuration Variables for POST" on page 5 for more information.

Sample POST Diagnostic Output

The following is partial sample output of POST testing for four online CPUs: CPU1, CPU3, CPU5, and CPU7. The CPUs CPU0, CPU2, CPU4, and CPU6 are offline. In the sample output, CPU1 is the master CPU, and the OpenBoot Diagnostics configuration variable `diag-level` is set to `max`. The CPU being tested is indicated by `1>`, `3>`, `5>`, or `7>` at the beginning of each status line.

```
@(#)OBP 4.0.45 2001/02/08 14:32 Sun Fire V890
Online: CPU0 CPU1 CPU2 CPU3 CPU4 CPU5 CPU6 CPU7*
Executing Power On SelfTest w/%o0 = 0000.0000.0003.1001

Calling POST w/%o0 0000.0000.0003.1001
1>@(#) Sun Fire V890 POST 1.2.45 2001/02/21 01:10
1>
1>Jump from OBP->POST.
1>System frequency is 150 MHz, CPU frequency 750
1>
1>Start selftest...
1>Offline CPU 0.
1>Offline CPU 2.
1>Reset Module with CPUs 2 0, both have been offlined.
1>Offline CPU 4.
1>Offline CPU 6.
1>Reset Module with CPUs 6 4, both have been offlined.
1>Init CPU
1>Scrub and Setup Ecache
1>     Size = 00000000.00800000...
1>Setup and Enable DMMU
1>Init Scan and I2C Devices
1>Creating Scan Database
1>INFO: Initializing MDR Chips...
1>INFO: Initializing DAR DTL bits ...
1>INFO: Initializing DCS DTL bits ...
1>INFO: Initializing All I2C Controllers and seg5_hp_en
1>Running scan ring integrity test
1>INFO: Ring 3 on BBC# 0 NOT Present or Shut OFF
1>INFO: Ring 5 on BBC# 0 NOT Present or Shut OFF
1>INFO: Ring 3 on BBC# 1 NOT Present or Shut OFF
1>INFO: Ring 5 on BBC# 1 NOT Present or Shut OFF
1>INFO: Disabling DAR-Err Circuitry ...
1>INFO: Setting Trip Temp of CPU 1 and 3 to 110C
1>INFO: Setting Trip Temp of CPU 5 and 7 to 110C
1>WED FEB  21 6:14:00 GMT 1
1>INFO: Disabling Cheetah-Err Circuitry ...
```

```
1>Setup DMMU Miss Handler
1>Probe and Setup Memory
1>INFO: 256MB Bank 0
1>INFO:No memory detected in Bank 1
1>INFO:No memory detected in Bank 2
1>INFO:No memory detected in Bank 3
1>Data Bitwalk on Master
1>     Test Bank 0.
1>Address Bitwalk on Master
1>INFO: Addr walk mem test on CPU 1 Bank 0: 00000010.00000000 to
00000010.10000000.
1>Set Mailbox
1>Move Memory Stack
1>     New memory location 00000010.00110000.
1>Post Data Region Scrub
1>Setup Final DMMU Entries
1>Post Image Region Scrub
1>Copy POST to Memory
1>Verifying checksum on copied image.
1>The Memory's CHECKSUM value is e92b.
1>The Memory's Content Size value is a91a0.
1>Success...  Checksum on Memory Validated.
3>Init CPU
5>Init CPU
7>Init CPU
3>Scrub and Setup Ecache
3>     Size = 00000000.00800000...
5>Scrub and Setup Ecache
5>     Size = 00000000.00800000...
7>Scrub and Setup Ecache
7>     Size = 00000000.00800000...
3>Setup and Enable DMMU
7>Setup and Enable DMMU
5>Setup and Enable DMMU
3>Setup DMMU Miss Handler
3>Probe and Setup Memory
3>WARNING:DIMM Failure detected in Bank 2
3>     DIMM 0 J7900 side 2 =   0MB.
3>     DIMM 1 J7901 side 2 =   0MB.
3>     DIMM 2 J8001 side 2 =   0MB.
3>     DIMM 3 J8000 side 2 =  64MB.
3>INFO: 256MB Bank 0
3>INFO:No memory detected in Bank 1
3>INFO:No memory detected in Bank 3
```

The remaining POST output would show the results of CPU and memory testing of CPU3, CPU5, and CPU7.

Sample POST Error Messages

If POST detects an error, it displays an error message indicating the failing part. If POST detects an error that prevents the system from booting, POST halts execution and returns control to OpenBoot firmware. The last message displayed by POST prior to the `ok` prompt indicates the part you need to replace.

The following is a sample error message for a failed test at DIMM J7900.

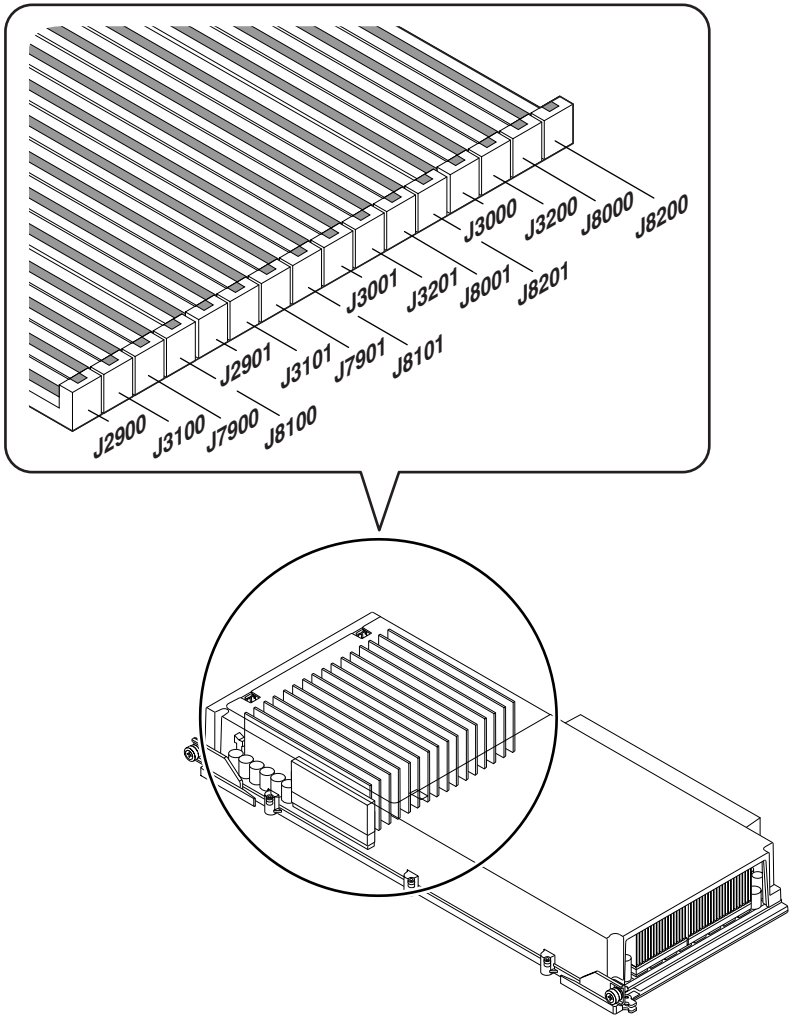
```
1>Data Bitwalk on Slave 3
1>                 Test Bank 0.
3>Bank 0 DIMM 0
3>ERROR: TEST = Bank 0 DIMM 0
3>H/W under test = CPU3 Bank 0 Dimm 0, J7900 side 1
3>MSG =
        *** Test Failed!! ***

3>END_ERROR

1>ERROR: TEST = Data Bitwalk on Slave 3
1>H/W under test = CPU3 Memory
1>MSG = ERROR:     miscompare on mem test!
                    Address: 00000030.001b0038
                    Expected: 00000000.00100000
                    Observed: 00000000.00000000
1>END_ERROR

1>ERROR: TEST = Data Bitwalk on Slave 3
1>H/W under test = CPU3 Memory
1>MSG =
        *** Test Failed!! ***
1>END_ERROR
```

As shown in the preceding error message, POST reports memory errors by indicating the location ("J" number) of the failing DIMM. Use the following diagram to identify the location of a failing DIMM according to J number.



Sample Summary of POST Results

POST results are saved across power cycles. To display the results of POST testing, type `.show-post-results` at the `ok` prompt. The following is a sample of `.show-post-results` output.

```
{2} ok .show-post-results
CPU0/Memory:   OK
CPU1/Memory:   OK
CPU2/Memory:   OK
CPU3/Memory:   OK
CPU4/Memory:   OK
CPU5/Memory:   OK
CPU6/Memory:   OK
CPU7/Memory:   OK
Schizo0:       OK
Schizo1:       OK
BBC0:          OK
BBC1:          OK
RIO:           OK
FCAL:          OK
GEM:           OK
SCSI:          OK
Ethernet:      OK
USB:           OK
RSC:           OK
GPTwo Slots:   OK
PCI Slots:     OK
ok
```

The failed status of a device is maintained until POST diagnostics are run again and the faulty device passes. If for some reason you want to override a failed status, set `diag-level` to `off` and rerun the diagnostics. With `diag-level` set to `off`, no tests are run, and POST indicates a passed status for all devices.

```
ok setenv diag-level off
ok reset-all
```

About OpenBoot Diagnostics

OpenBoot Diagnostics code resides in the OpenBoot PROM on the system I/O board. OpenBoot Diagnostics can detect and isolate errors in the following system components:

- Motherboard and I/O board
- Disk drives and FC-AL disk backplanes
- Any PCI card that contains an on-board self-test

OpenBoot Diagnostics also tests the following I/O interfaces:

- PCI
- FC-AL
- Gigabit Ethernet
- Fast Ethernet
- Serial
- USB
- RSC

You can run OpenBoot Diagnostics tests in the following ways:

- From the `ok` prompt. See "OpenBoot `ok` Prompt Commands" on page 22.
- From the OpenBoot Diagnostics menu. See "OpenBoot Diagnostics Menu" on page 26 and "Running OpenBoot Diagnostics" on page 31.
- Automatically after certain events. See "Running OpenBoot Diagnostics" on page 31.

OpenBoot Diagnostics reports test results through detailed diagnostic and error messages. See "Sample OpenBoot Diagnostics Error Messages" on page 37 for information about error messages.

When executed automatically, OpenBoot Diagnostics displays status and error messages through a `tip` connection or a local ASCII terminal attached to the system's serial port A (`ttya`). You can also redirect OpenBoot Diagnostics messages to a remote RSC console. If you redirect output to an RSC console, you cannot display OpenBoot Diagnostics results locally. For more information about RSC, see "Displaying OpenBoot Diagnostics Results" on page 36.

When executed interactively from the `ok` prompt or the OpenBoot Diagnostics menu, OpenBoot Diagnostics displays status and error messages on any system console, including a local graphics console.

The OpenBoot firmware provides diagnostic configuration variables that you can set to control the operation of the OpenBoot Diagnostics tests. For information about the configuration variables, see "OpenBoot Configuration Variables for OpenBoot Diagnostics" on page 18.

OpenBoot Configuration Variables for OpenBoot Diagnostics

The following table lists and describes the OpenBoot Diagnostics configuration variables that control the operation of OpenBoot Diagnostics.

Note – Both OpenBoot Diagnostics and POST use the settings of the configuration variables `diag-level`, `diag-switch?`, and `diag-trigger`. Changing the values of these variables will affect both POST and OpenBoot Diagnostics operation. See "OpenBoot Configuration Variables for POST" on page 5 for a complete listing and description of the OpenBoot configuration variables that control POST testing.

Variable	Setting or Keyword	Description	Default
`diag-level`		Determines the level of testing executed. **Note:** If `diag-level` is set to `menus` (for POST interactive operation), OpenBoot Diagnostics runs the default level (`min`) of testing. See "OpenBoot Configuration Variables for POST" on page 5 for more information about the `menus` setting.	`max`
	`off`	Performs no OpenBoot Diagnostics testing. **Note:** If `diag-level` is set to `off`, OpenBoot Diagnostics returns a passed status for all self-tests, but no testing is performed.	
	`min`	Performs minimal testing of core device functions.	
	`max`	Performs maximum testing of device functions.	
`diag-passes`	*n*	Specifies the number of consecutive executions of OpenBoot Diagnostics tests that are run from the OpenBoot Diagnostics menu. The maximum value for `diag-passes` is >1,000,000. **Note:** The variable `diag-passes` has no effect outside the OpenBoot Diagnostics Menu. See "OpenBoot `ok` Prompt Commands" on page 22.	`1`

Variable	Setting or Keyword	Description	Default
diag-script		Determines which OpenBoot Diagnostics tests are run automatically after the reset event specified by the variable diag-trigger.	normal
	normal	Tests all the devices shipped with a base system.	
	all	Executes all available self-tests, including tests on plug-in cards. (Same as executing test-all from the ok prompt.)	
	none	No diagnostic self-tests are run.	
diag-switch?		Controls diagnostic execution in normal mode.	false
	true	Diagnostics are *only* executed on power-on reset events, but the level of test coverage, verbosity, and output is determined by user-defined settings.	
	false	Diagnostics are executed upon next system reset, but only for those class of reset events specified by the OpenBoot configuration variable diag-trigger. The level of test coverage, verbosity, and output is determined by user-defined settings.	
diag-trigger		Specifies the class of reset event that causes diagnostics to run automatically. **Note**: Both POST and OpenBoot Diagnostics run at the specified reset event if the variable diag-script is set to normal or all. If diag-script is set to none, only POST runs.	power-on-reset error-reset
	none	Diagnostic tests are not executed.	
	error-reset	Reset that is caused by certain hardware error events such as RED State Exception Reset, Watchdog Resets, Software-Instruction Reset, or Hardware Fatal Reset.	
	power-on-reset	Reset that is caused by power cycling the system.	
	user-reset	Reset that is initiated by an operating system panic or by userinitiated commands from OpenBoot (reset-all or boot) or from Solaris (reboot, shutdown, or init).	
	all-resets	Any kind of reset.	
service-mode		Diagnostics are executed at Sun-specified levels, overriding but preserving user settings.	false

Variable	Setting or Keyword	Description	Default
	false	Normal mode, unless overridden by the panel keyswitch. Diagnostics execution depends entirely on the settings of diag-switch? and other user-defined OpenBoot configuration variables.	
	true	Service mode. **Note**: If the panel keyswitch is in the Diagnostics position, the system will boot in service mode even if the service-mode? variable is false.	
test-args		Customizes OpenBoot Diagnostics tests. Allows a text string of reserved keywords (separated by commas) to be specified in the following ways: • As an argument to the test command at the ok prompt • As an OpenBoot variable to the setenv command at the ok or obdiag> prompt The following are the reserved keywords for the variable test-args:	Empty string
	bist	Invokes built-in self-test (BIST) on external and peripheral devices.	
	debug	Displays all debug messages.	
	hotplug	Enables hot-plug controller tests. (Power cycles PCI slots.) **Warning:** After the hot-plug test, the PCI cards in the slots tested are not usable until you reset the system.	
	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	Suppresses messages announcing the name of every test run from the OpenBoot Diagnostics menu commands. (This keyword has no effect on status messages of tests run from the ok prompt.)	
	subtests	Displays name of each subtest that is called.	
	verbose	Displays detailed messages of progression of all tests.	

Variable	Setting or Keyword	Description	Default
	`callers=`*N*	Displays backtrace of *N* callers when an error occurs. • `callers=0` displays backtrace of all callers on error.	
	`errors=`*N*	Continues executing the test until *N* errors are encountered. • `errors=0` displays all error reports without terminating testing.	
`verbosity`		Controls the amount and detail of OpenBoot, POST, and OpenBoot Diagnostics output.	`none`
	`none`	Only error and fatal messages are displayed on the system console. Banner is not displayed. **Note**: Problems in systems with `verbosity` set to `none` might be deemed not diagnosable, rendering the system unserviceable by Sun.	
	`min`	Notice, error, warning, and fatal messages are displayed on the system console. Transitional states and banner are also displayed.	
	`normal`	Summary progress and operational messages are displayed on the system console in addition to the messages displayed by the `min` setting. The work-in-progress indicator shows the status and progress of the boot sequence.	
	`max`	Detailed progress and operational messages are displayed on the system console in addition to the messages displayed by the `min` and `normal` settings.	

To display the current values of all OpenBoot configuration variables, use the `printenv` command at the `ok` prompt without specifying a variable name. To display the current values of the OpenBoot Diagnostics configuration variables, use the `printenvs` menu command at the `obdiag>` prompt. For more details, see "OpenBoot Diagnostics Menu Commands" on page 27.

```
obdiag> printenvs
Variable Name           Value           Default Value

diag-switch?            true            false
diag-level              min             min
test-args               subtests
diag-passes             10              1

obdiag>
```

To set or change the value of a diagnostic configuration variable, use the `setenv` command at the `ok` prompt or at the `obdiag>` prompt. See "OpenBoot Diagnostics Menu Commands" on page 27 for more information.

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

OpenBoot `ok` Prompt Commands

OpenBoot Diagnostics detects any device that has a self-test that supports the OpenBoot standard. These devices can include both components of the basic system and any optional device with a self-test that supports the standard. Any of these devices can be tested from the `ok` prompt using the `test` or `test-all` commands. The `test` and `test-all` commands allow you to specify a particular device for testing. For more information about performing tests using the `ok` prompt commands, see "`test` Command" on page 23 and "`test-all` Command" on page 25.

Note – You should run OpenBoot Diagnostics tests at the `ok` prompt only after a power-on or system reset. You cannot run OpenBoot Diagnostics reliably after halting the operating system or aborting the operating system with the Stop-A keyboard command (or an equivalent abort key sequence). Therefore, in order to access the `ok` prompt and run OpenBoot Diagnostics, you must set the OpenBoot configuration variable `auto-boot?` to `false` and reset the system. For the detailed procedure, see "Initiating OpenBoot Diagnostics" on page 35.

`test` Command

The `test` command enables you to test an individual device. At the `ok` prompt, type `test` and the full path name or device alias of the device, as shown in the following example:

```
ok test /pci@9,700000/ebus@1/flashprom@0,0
```

To display the list of system device aliases, type `devalias` at the `ok` prompt.

```
{0} ok devalias
cdrom                    /pci@8,700000/ide@1/cdrom@0,0:f
ide                      /pci@8,700000/ide@1
disk                     /pci@8,600000/SUNW,qlc@2/fp@0,0/disk@0,0
disk0                    /pci@8,600000/SUNW,qlc@2/fp@0,0/disk@0,0
disk1                    /pci@8,600000/SUNW,qlc@2/fp@0,0/disk@1,0
disk2                    /pci@8,600000/SUNW,qlc@2/fp@0,0/disk@2,0
disk3                    /pci@8,600000/SUNW,qlc@2/fp@0,0/disk@3,0
disk4                    /pci@8,600000/SUNW,qlc@2/fp@0,0/disk@4,0
disk5                    /pci@8,600000/SUNW,qlc@2/fp@0,0/disk@5,0
disk6                    /pci@8,600000/SUNW,qlc@2/fp@0,0/disk@8,0
disk7                    /pci@8,600000/SUNW,qlc@2/fp@0,0/disk@9,0
disk8                    /pci@8,600000/SUNW,qlc@2/fp@0,0/disk@a,0
disk9                    /pci@8,600000/SUNW,qlc@2/fp@0,0/disk@b,0
disk10                   /pci@8,600000/SUNW,qlc@2/fp@0,0/disk@c,0
disk11                   /pci@8,600000/SUNW,qlc@2/fp@0,0/disk@d,0
scsi                     /pci@8,600000/SUNW,qlc@2
net                      /pci@9,700000/network@1,1
gem                      /pci@8,600000/network@1
flash                    /pci@9,700000/ebus@1/flashprom@0,0
idprom                   /pci@9,700000/ebus@1/i2c@1,500030/idprom@0,a0
nvram                    /pci@9,700000/ebus@1/i2c@1,500030/nvram@0,a0
i2c3                     /pci@9,700000/ebus@1/i2c@1,500030
i2c2                     /pci@9,700000/ebus@1/i2c@1,50002e
bbc1                     /pci@9,700000/ebus@1/bbc@1,500000
i2c1                     /pci@9,700000/ebus@1/i2c@1,30
i2c0                     /pci@9,700000/ebus@1/i2c@1,2e
bbc0                     /pci@9,700000/ebus@1/bbc@1,0
rsc-console              /pci@9,700000/ebus@1/rsc-console@1,3083f8
rsc-control              /pci@9,700000/ebus@1/rsc-control@1,3062f8
ttyb                     /pci@9,700000/ebus@1/serial@1,400000:b
ttya                     /pci@9,700000/ebus@1/serial@1,400000:a
pci9b                    /pci@9,700000
pci9a                    /pci@9,600000
pci8b                    /pci@8,700000
pci8a                    /pci@8,600000
ebus                     /pci@9,700000/ebus@1
```

You can use `test-args` keywords with the `test` command to fine-tune the execution of the test. See "OpenBoot Configuration Variables for OpenBoot Diagnostics" on page 18 for more information about the `test-args` options. The following is an example of using the `test-args` keywords `loopback` and `verbose` with the `test` command:

```
ok test /pci@9,700000/network@1:test-args={loopback,verbose}
```

`test-all` Command

When no device path is specified, the `test-all` command tests all devices with self-tests as detected by OpenBoot Diagnostics:

```
ok test-all
Testing /pci@9,700000/usb@1,3
Testing /pci@9,700000/network@1,1
Testing /pci@9,700000/ebus@1
Testing /pci@9,700000/ebus@1/serial@1,400000
Testing /pci@9,700000/ebus@1/rsc-control@1,3062f8
Testing /pci@9,700000/ebus@1/pmc@1,300700
Testing /pci@9,700000/ebus@1/gpio@1,300600
Testing /pci@9,700000/ebus@1/rtc@1,300070
Testing /pci@9,700000/ebus@1/i2c@1,500030
Testing /pci@9,700000/ebus@1/i2c@1,50002e
Testing /pci@9,700000/ebus@1/bbc@1,500000
Testing /pci@9,700000/ebus@1/i2c@1,30
Testing /pci@9,700000/ebus@1/i2c@1,2e
Testing /pci@9,700000/ebus@1/bbc@1,0
Testing /pci@9,700000/ebus@1/flashprom@0,0
Testing /pci@9,700000/ebus@1/i2c@1,30/hotplug-controller@0,ec
Testing /pci@9,700000/ebus@1/i2c@1,30/hotplug-controller@0,e8
Testing /pci@9,700000/ebus@1/i2c@1,30/hotplug-controller@0,e6
Testing /pci@9,700000/ebus@1/i2c@1,30/hotplug-controller@0,e2
Testing /pci@9,700000/ebus@1/i2c@1,30/controller@0,1a
Testing /pci@9,700000/ebus@1/i2c@1,30/controller@0,16
Testing /pci@8,600000/SUNW,qlc@2
Testing /pci@8,600000/network@1
Testing /pci@8,700000/ide@1
```

OpenBoot Diagnostics Menu

The OpenBoot Diagnostics menu is displayed when you issue the `obdiag` command at the `ok` prompt. OpenBoot Diagnostics detects each device with a self-test and displays that device name in the OpenBoot Diagnostics menu. The OpenBoot Diagnostics menu always includes the devices of the basic system. These devices include: `bbc`, `controller`, `ebus`, `flashprom`, `gpio`, `hotplugcontroller`, `ide`, `i2c`, `network`, `pmc`, `rsc-control`, `rtc`, `scsi`, `serial`, and `usb`.

If an optional plug-in device has a self-test that supports the OpenBoot standard, the OpenBoot Diagnostics menu also includes that device as one of the menu entries. Therefore, the menu entries may vary from system to system, depending on the optional devices installed in the system.

You invoke the OpenBoot Diagnostics menu by typing `obdiag` at the `ok` prompt. A sample OpenBoot Diagnostics menu is shown below.

```
ok obdiag
 ____________________________________________________________________________
|                                o b d i a g                                 |
|____________________________ ____________________________ ____________________|
|                            |                            |                    |
|  1 SUNW,qlc@2              |  2 bbc@1,0                 |  3 bbc@1,500000    |
|  4 controller@0,16         |  5 controller@0,1a         |  6 ebus@1          |
|  7 flashprom@0,0           |  8 gpio@1,300600           |  9 hotplug-controller@0,|
| 10 hotplug-controller@0,   | 11 hotplug-controller@0,   | 12 hotplug-controller@0,|
| 13 i2c@1,2e                | 14 i2c@1,30                | 15 i2c@1,50002e    |
| 16 i2c@1,500030            | 17 ide@1                   | 18 network@1       |
| 19 network@1,1             | 20 pmc@1,300700            | 21 rsc-control@1,3062f8 |
| 22 rtc@1,300070            | 23 serial@1,400000         | 24 usb@1,3         |
|____________________________|____________________________|____________________|
|       Commands: test test-all except help what setenv set-default exit     |
|____________________________________________________________________________|
|                    diag-passes=1 diag-level=min test-args=                 |
|____________________________________________________________________________|

obdiag>
```

For information about each OpenBoot Diagnostics test, see "OpenBoot Diagnostics Test Descriptions" on page 27. For a description of the interactive commands that allow you to run OpenBoot Diagnostics from the `obdiag>` prompt, see "OpenBoot Diagnostics Menu Commands" on page 27.

OpenBoot Diagnostics Menu Commands

The following table describes the interactive OpenBoot Diagnostics menu commands that are available at the `obdiag>` prompt.

Command	Description
`exit`	Exits the OpenBoot Diagnostics menu and returns to the `ok` prompt.
`help`	Displays a brief description of each OpenBoot Diagnostics menu command and OpenBoot configuration variable.
`printenvs`	Displays the current value of diagnostics-related OpenBoot configuration variables. (See "OpenBoot Configuration Variables for OpenBoot Diagnostics" on page 18 for information about the configuration variable values.)
`setenv` *variable-value*	Sets the value for an OpenBoot configuration variable. (See "OpenBoot Configuration Variables for OpenBoot Diagnostics" on page 18 for information about the configuration variable values.)
`test-all`	Tests all devices displayed in the OpenBoot Diagnostics menu. **Note:** Unlike the `test-all` command at the `ok` prompt, the `test-all` menu command at the `obdiag>` prompt does not allow you to specify a device path name.
`versions`	Displays the version, last modified date, and manufacturer of each self-test and the OpenBoot Diagnostics menu and library.
`test #,#,`	Tests only the device or devices identified by the menu entry number (#) in the command line. Specify individual tests, separated by commas. (Example: `obdiag>` **`test 7,10`**)
`except #,#,`	Tests all devices in the OpenBoot Diagnostics menu except those identified in the list. (Example: `obdiag>` **`except 3,5,10`**)
`what #,#,`	Displays selected properties of the devices identified by the menu entry number (#) in the command line. The information provided varies according to device type.

OpenBoot Diagnostics Test Descriptions

OpenBoot Diagnostics provides comprehensive diagnostic testing for the I/O subsystem, I²C subsystem, and other hardware devices. Tests available through OpenBoot Diagnostics are:

- Core tests, which exercise parts of the basic system
- On-board self-tests, which exercise optional devices such as PCI cards

Note – For maximum testing of each device, set the `diag-level` variable to `max`; for limited testing, set `diag-level` to `min`. For some devices, the testing is the same at both the `min` and `max` settings.

The following table lists the devices provided with a typical system and describes the self-test of each device. The table provides the device path name, a brief description of the device's self-test, and any special considerations involved in running the test.

Note – The `test-args` keywords `verbose`, `subtests`, `debug`, `errors=`*N*, and `callers=`*N* apply to all self-tests.

Device	Description of Device Self-Test	Special Considerations
`bbc@1,0` `bbc@1,500000`	Tests all writable registers in the boot bus controller and then verifies that at least one processor has boot bus access.	
`controller@0,16` `controller@0,1a`	Executes the tests in the base FC_AL backplane firmware and SSC_100 SES controllers.	
`controller@0,1c` `controller@0,1e`	Executes the tests in the expansion FC-AL backplane firmware and SSC-100 SES controllers.	Only available on systems equipped with optional expansion FC-AL backplane.
`ebus@1`	Tests the PCI configuration registers, DMA control registers, and ebus mode registers. Tests DMA controller functions.	
`flashprom@0,0`	Performs a checksum of the flash PROM containing the OpenBoot firmware.	
`gpio@1,300600`	Tests the registers of the super I/O subsystem.	

`hotplugcontroller@0,e2`
`hotplugcontroller@0,e6`
`hotplugcontroller@0,e8`
`hotplugcontroller@0,ec`
	Performs hot-plug test of PCI slots.
	
	Warning: After the hot-plug test, the PCI cards in the slots tested are not usable until you reset the system.
		To run hot-plug tests, the `test-args` keyword `hotplug` must be specified.

Device	Description of Device Self-Test	Special Considerations
i2c@1,2e i2c@1,30 i2c@1,50002e i2c@1,500030	Tests the devices (temperature sensors, fans, power supplies, system fault LEDs, thermal fault LEDs, and front panel keyswitch) monitored by the I^2C environmental monitoring bus.	
ide@1	Tests the on-board IDE controller and IDE bus subsystem for internal removable media devices. Checks associated registers and performs a DMA transfer.	You must specify the variable `test-args` keywords `media` and `bist`.
network@1,1	Tests the on-board Fast Ethernet logic, including internal and external loopback tests.	To run the external loopback test on the TPE port, you must have a TPE loopback connector attached to the TPE port and specify the `test-args` keyword `loopback`. The Sun part number for the TPE loopback connector is 501-2965-01.
network@1	Tests the on-board Gigabit Ethernet (GBE) logic, including internal and external loopback tests.	To run the external loopback test on the GBE port, you must have a GBE loopback connector attached to the GBE port and specify the `test-args` keyword `loopback`. This connector consists of looping back one end of the optical connector to the other end using any standard optical cable.
pmc@1,300700	Tests the registers of the power management controller.	
SUNW,qlc@2	Tests the registers of the on-board FC-AL controller and FC-AL subsystem (Loop A).	

Device	Description of Device Self-Test	Special Considerations
rsc-control@1,3062f8	Tests RSC hardware, including RSC serial and Ethernet ports.	To run external loopback tests on the RSC Ethernet port: • Variable diag-level must be set to max. • Variable test-args string must specify the keyword loopback. • RSC Ethernet port must be connected to a 10-Mbyte hub. To run external loopback tests on the RSC serial port: • Variable diag-level must be set to max. • Variable test-args string must specify the keyword loopback.
rtc@1,300070	Tests the registers of the real-time clock and then tests the interrupt rates.	To test the ability to enable or disable the daylight savings time feature, the variable diag-level must be set to max.
scsi@1	Tests optional SCSI controllers and SCSI bus subsystem for optional removable media devices. Checks associated registers and performs a DMA transfer.	You must specify the variable test-args keywords media and bist.
serial@1,400000	Tests all possible baud rates supported by the ttya and ttyb serial lines and performs an internal and external loopback test on each line at each speed.	If a serial line is being used by an I/O device, that line will not be tested. To run the external loopback test on the serial lines: • Variable test-args must specify the keyword loopback. • You must have a loopback connector attached to each serial port with the ttya line transmitting while the ttyb line is looped back. The Sun part number for the serial loopback connector is 501-4205-01.
usb@1,3	Tests the writable registers of the USB open host controller.	

Additional testing may be performed if your configuration includes an optional device that has an on-board self-test that supports the OpenBoot standard. Such optional devices include PCI interface cards that support parallel communication lines, audio devices, or any other device that is IEEE 1275 compatible and provides a method named "selftest." Examples of optional devices are:

- `fdthree` – Self-test for this device tests the control logic of a diskette drive and the operation of the drive. (A formatted diskette must be inserted into the diskette drive.)
- `SUNW,CS4231`– Self-test for this device verifies that an audio PCI card is present and tests associated registers. Specifying the keyword `loopback` for the `test-args` OpenBoot Diagnostics configuration variable enables the following tests: external line-in/line-out loopback tests, external speaker tone tests, and external microphone/headphone loopback tests.

Running OpenBoot Diagnostics

When you power on the system, OpenBoot Diagnostics runs automatically under either of the following conditions:

- The front panel keyswitch is set to the Diagnostics position and both of the OpenBoot configuration variables `diag-level` and `diag-script` are set to their default values (or to any valid setting other than `off`).
- The OpenBoot configuration variable `diag-switch?` is set to `true` and all three of the variables `diag-level`, `diag-trigger`, and `diag-script` are set to their default values (or to any valid setting other than `off`).

Note – The default value for `diag-switch?` is `false`. Therefore, if all OpenBoot configuration variables are set to their default values, OpenBoot Diagnostics does not run automatically unless the keyswitch is set to the Diagnostics position or the `service-mode?` variable is set to `true`. For maximum test coverage, set the `diag-level` variable to `max` prior to starting OpenBoot Diagnostics.

You can configure OpenBoot Diagnostics to run automatically after specific types of reset events by setting the values of the variables `diag-switch?` and `diag-trigger`, as shown in the following table. Note that `diag-level` and `diag-script` must be set to any valid value other than `none`. For more information, see "OpenBoot Configuration Variables for OpenBoot Diagnostics" on page 18.

Reset Event	OpenBoot Diagnostics Runs Automatically If...
Any power-on reset, including RSC-initiated power-on resets	The front panel keyswitch is set to the Diagnostics position OR `diag-switch?` is set to `true` and `diag-trigger` is set to any setting other than `none`
Any automatic reset triggered by a hardware error, including all operating system panics and watchdog reset events	`diag-switch?` is set to `true` and `diag-trigger` is set to `error-reset` or `soft-reset`
Any user-initiated reset event	`diag-switch?` is set to `true` and `diag-trigger` is set to `soft-reset`

The setting for `diag-script` determines which tests are run at the reset event specified by `diag-trigger`. Valid settings for `diag-script` are:

- `normal` – Tests all devices shipped with a base system.
- `all` – Executes all available self-tests, including tests on plug-in cards.
- `none` – No diagnostic self-tests are run.

See "OpenBoot Configuration Variables for OpenBoot Diagnostics" on page 18 for information about the settings for `diag-script`.

The following sample output shows the results of OpenBoot Diagnostics tests when the variable `diag-level` is set to `max`, `diag-script` is set to `normal`, and no `test-args` keywords are specified.

```
Running diagnostics script obdiag/normal

Testing /pci@8,600000/network@1
Testing /pci@8,600000/SUNW,qlc@2
Testing /pci@9,700000/ebus@1/i2c@1,2e
Testing /pci@9,700000/ebus@1/i2c@1,30
Testing /pci@9,700000/ebus@1/i2c@1,50002e
Testing /pci@9,700000/ebus@1/i2c@1,500030
Testing /pci@9,700000/ebus@1/bbc@1,0
Testing /pci@9,700000/ebus@1/bbc@1,500000
Testing /pci@8,700000/ide@1
Testing /pci@9,700000/network@1,1
Testing /pci@9,700000/usb@1,3
Testing /pci@9,700000/ebus@1/gpio@1,300600
Testing /pci@9,700000/ebus@1/pmc@1,300700
Testing /pci@9,700000/ebus@1/rtc@1,300070
```

OpenBoot Diagnostics runs automatically, without operator intervention, under the conditions described previously. However, you can also run OpenBoot Diagnostics in an interactive mode and specify which tests you want to perform. OpenBoot Diagnostics tests can be executed interactively in the following ways:

- From the `ok` prompt, you can use either the `test` or `test-all` command to test a particular device. See "`test` Command" on page 23 and "`test-all` Command" on page 25.
- From the `obdiag>` prompt, you can use the OpenBoot Diagnostics menu commands to execute the self-tests of the devices included in the OpenBoot Diagnostics menu.

How to Run OpenBoot Diagnostics

The following procedure describes how to run OpenBoot Diagnostics interactively from the `obdiag>` prompt. There are two parts to this procedure:

- "Initiating OpenBoot Diagnostics" on page 35
- "Displaying OpenBoot Diagnostics Results" on page 36

Following this procedure is:

- "Sample OpenBoot Diagnostics Error Messages" on page 37

Before You Begin

You need to set up a way of viewing OpenBoot Diagnostics error and diagnostic messages if your server is configured without a system console. Use the following guidelines to set up a way of displaying the messages for your particular installation:

- If you are running OpenBoot Diagnostics interactively, you can:
  - Connect a local graphics console or an alphanumeric terminal to the Sun Fire V890 server. See the *Sun Fire V890 Server Owner's Guide*.
  - Establish a `tip` connection from another Sun system. See the *OpenBoot 4.x Command Reference Manual*.
  - Set up an RSC console and direct output to the RSC console. For more information, see "About Sun Remote System Control Software" on page 46.
- If OpenBoot Diagnostics will be running automatically after a power-on or reset event, you can:
  - Connect an alphanumeric terminal. See the *Sun Fire V890 Server Owner's Guide* for instructions. (You cannot view diagnostics messages at a graphics console when OpenBoot Diagnostics is running automatically.)
  - Establish a `tip` connection from another Sun system. See the *OpenBoot 4.x Command Reference Manual*.
  - Set up an RSC console and direct output to the RSC console. For more information, see "About Sun Remote System Control Software" on page 46.

Note – When executed automatically, OpenBoot Diagnostics output displays locally on an attached terminal or through a `tip` connection. However, if diagnostics output is redirected to an RSC console, the output will not display locally until it is directed back to the local terminal or `tip` connection. For information about directing OpenBoot Diagnostics output to an RSC console or to a local terminal or `tip` connection, see the *Sun Remote System Control (RSC) User's Guide* and "Displaying OpenBoot Diagnostics Results" on page 36.

Initiating OpenBoot Diagnostics

You should run OpenBoot Diagnostics tests interactively only after a power-on or system reset. You cannot run OpenBoot Diagnostics reliably after halting the operating system or aborting the operating system with the Stop-A keyboard command (or an equivalent abort key sequence). Therefore, in order to access the `ok` prompt and run OpenBoot Diagnostics, you must set the OpenBoot configuration variable `auto-boot?` to `false` and reset the system.

Perform the following steps to set the configuration variable `auto-boot?` and to run the OpenBoot Diagnostics tests interactively.

1. **Access the `ok` prompt.**

   To access the `ok` prompt:

   - On a Sun keyboard, hold down the Stop key and press A.
   - On a terminal keyboard, press the Break key.
   - Type ~# in a `tip` window.

   The `ok` prompt is displayed.

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

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

3. **Reset or power cycle the system, type:**

```
ok reset-all
```

4. **When the `ok` prompt appears, invoke OpenBoot Diagnostics, type:**

```
ok obdiag
```

   The OpenBoot Diagnostics menu appears.

5. ***(Optional)* When the OpenBoot Diagnostics menu and `obdiag>` prompt appear, set the configuration variables.**

   See "OpenBoot Configuration Variables for OpenBoot Diagnostics" on page 18 for information about the variable values.

   The following example shows how to set the value for the variable `diag-level`, which specifies the level of testing performed:

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

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

6. **To execute one or more tests, enter the appropriate OpenBoot Diagnostics menu command and test numbers at the `obdiag>` prompt.**

   The following example shows the `except` command, which allows you to execute all tests except those tests you specify in the command:

```
obdiag> except 1,4
```

   For command usage and descriptions, see "OpenBoot Diagnostics Menu Commands" on page 27.

   For information about the OpenBoot Diagnostics tests, see "OpenBoot Diagnostics Menu" on page 26 and "OpenBoot Diagnostics Test Descriptions" on page 27.

Displaying OpenBoot Diagnostics Results

By default, when you run OpenBoot Diagnostics interactively, the output displays locally on the system console. You can redirect OpenBoot Diagnostics output to display remotely on an RSC console.

To redirect output to an RSC console:, follow these steps

1. **Type the following commands at the system `ok` prompt:**

```
ok diag-console rsc
ok setenv input-device rsc-console
ok setenv output-device rsc-console
```

2. **To cause the changes to take effect, power cycle the system, or type:**

```
ok reset-all
```

If you redirect OpenBoot Diagnostics output to an RSC console, the output will not display on the system console. To redirect OpenBoot Diagnostics output to the local system console or to a `tip` connection, issue the `diag-console` command as shown in the following example:

```
ok diag-console ttya
ok reset-all
```

See the *Sun Remote System Control (RSC) User's Guide* for more information about redirecting output to an RSC console.

Sample OpenBoot Diagnostics Error Messages

Using the OpenBoot configuration variable `test-args`, you can specify keywords to set reporting controls for diagnostic and error messages:

- `debug` – Provides all debug messages.
- `silent` – Suppresses display of test name.
- `verbose` – Provides detailed test status messages.
- `callers=`*N* – Sets the number of backtrace callers reported.
- `errors=`*N* – Sets the number of errors reported before testing is terminated.

See "OpenBoot Configuration Variables for OpenBoot Diagnostics" on page 18 and "Error Messages" on page 50 for additional information about the `test-args` variable. The following is an example of how to use the variable `test-args`.

```
ok setenv test-args verbose,debug,errors=0
```

OpenBoot Diagnostics reports errors in a standard format. The following output shows the `test` command for the FC-AL subsystem issued from the `obdiag>` prompt and a sample error message.

```
obdiag> test 1
Testing /pci@8,60000/SUNW,qlc@2

ERROR   : No command DMA interrupt
DEVICE  : /pci@8,60000/SUNW,qlc@2
SUBTEST : selftest:loop-host-fifo-host
CALLERS : loop-host-fifo-host
MACHINE : Sun Fire V890
SERIAL# : 12980798
DATE    : 04/30/2001 16:05:39 GMT

/pci@8,600000/SUNW,qlc@2 selftest failed, return code = 1
ok
```

About SunVTS Software

SunVTS, the Sun Validation Test Suite, is an online diagnostic tool and system exerciser for verifying the configuration and functionality of hardware controllers, devices, and platforms.

SunVTS software lets you view and control a testing session over modem lines or over a network. Using a remote system, you can view the progress of a SunVTS testing session, change testing options, and control all testing features of another system on the network.

SunVTS User Interfaces

SunVTS software provides the following user interfaces:

- Command-line interface
- TTY interface
- Graphical user interface (GUI) that runs within a windowed desktop environment

You can run SunVTS software from any one of its interfaces.

For More Information

The following documents provide information about SunVTS software. They are available on the Supplement CD for your specific Solaris release and on the web at `http://docs.sun.com`.

- *SunVTS User's Guide*

This document describes the SunVTS environment, including how to start and control the various user interfaces.

- *SunVTS Test Reference Manual*

This document describes each SunVTS test, the various test options, and command-line arguments.

- *SunVTS Quick Reference Card*

This card gives an overview of the main features of the SunVTS graphical user interface.

- *SunVTS Documentation Supplement*

This document is a supplement to the SunVTS documentation and describes new features, tests, and test enhancements that are developed in the SunVTS 5.1 Patch Set releases.

How to Check Whether SunVTS Software Is Installed

SunVTS software is an optional package that may or may not have been loaded when your system software was installed.

Before You Begin

This procedure assumes that the Solaris operating environment is running on the Sun Fire V890 server, and that you have access to the Solaris command line. For more information, see the *Sun Fire V890 Server Owner's Guide*

What to Do

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

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

   - If SunVTS software is loaded, information about the packages is displayed.
   - If SunVTS software is not loaded, you will 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 Operating System 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 the Solaris 8 Operating System.

   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 the Solaris 8 Operating System, 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℠ 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.

What Next

For installation information, refer to the *SunVTS Users Guide*, the appropriate Solaris documentation, and the `pkgadd` man page.

How to Run SunVTS Software

Before You Begin

If your system passes POST and OpenBoot Diagnostics testing and boots the operating system, yet does not function correctly, you can use SunVTS software to run additional tests. These tests verify the configuration and functionality of most hardware controllers and devices.

You will need superuser (root) access to run SunVTS tests.

What to Do

This procedure assumes you will test the server remotely by running a SunVTS session from a remote system using the SunVTS graphical interface. For information about the SunVTS interfaces and options, see the *SunVTS User's Guide.*

1. **Use the `xhost` command to give the Sun Fire V890 server access to the remote display.**

   On the remote system that will be running the SunVTS graphical interface, type:

   ```
   % /usr/openwin/bin/xhost + server-hostname
   ```

   Substitute the host name of the Sun Fire V890 server for *server-hostname*.

2. **Log in to the Sun Fire V890 server as superuser (root).**

   ```
   % rlogin server-hostname
   ```

3. **Check whether SunVTS software is loaded on the Sun Fire V890 server.**

   SunVTS is an optional package that may or may not have been loaded when the server software was installed. For more information, see "How to Check Whether SunVTS Software Is Installed" on page 40.

4. **To start the SunVTS software, type:**

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

Substitute the name of the system you are using for *system-hostname*. Note that `/opt/SUNWvts/bin` is the default directory for SunVTS software. If you have installed SunVTS software in a different directory, use the appropriate path instead.

5. **Fine-tune your testing session by selecting only the tests you want to run.**

   On the Test Selection panel, click to select and deselect tests. (A check mark in the box indicates the item is selected.) The following table lists and describes useful tests to run on the Sun Fire V890 server.

SunVTS Test	Description
`cdtest` `dvdtest`	Tests the DVD/CD-ROM drive by reading the disc and verifying the DVD/CD table of contents (TOC), if it exists
`cmttest`	Tests multi-core CPU
`cputest`	Tests the CPU
`disktest`	Verifies the internal SCSI bus and FC-AL disk drives
`dpmtest`	Verifies local FC-AL disk drives
`env5test` `i2ctest`	Tests the I^2C environment control system including all fans, all LEDs, front panel keyswitch, power supplies, and temperature sensors
`fputest`	Checks the floating-point unit
`iutest`	Tests the Integer Unit of the CPU
`l1dcachetest`	Tests the level 1 D cache on the CPU
`l2sramtest`	Tests the level 2 cache of the CPU
`m64test`	Tests the PCI graphics card
`mptest`	Verifies multiprocessor features (for systems with more than one processor)
`nettest` `netlbtest`	Checks all the hardware associated with networking (for example, Ethernet, token ring, quad Ethernet, fiber optic, 100-Mbit per second Ethernet, Gigabit Ethernet devices)
`pmem`	Tests the physical memory (read only)
`qlctest`	Tests the FC-AL controller
`ramtest`	Stress tests memory modules (RAM)
`sptest`	Tests the system's on-board serial ports
`ssptest`	Verifies the RSC functionality, including RSC Ethernet and serial ports, I^2C, and Flash RAM
`systest`	Stress tests both memory and CPUs

SunVTS Test	Description
`tapetest`	Tests the various Sun tape devices
`usbkbtest`	Tests the keyboard
`vmem`	Tests virtual memory (a combination of the swap partition and the physical memory)

SunVTS Results

If SunVTS tests indicate an impaired or defective part, see the replacement procedures in the *Sun Fire V890 Server Service Manual*.

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, see the manuals that accompany SunVTS software. These are listed in the section "Related Documentation" on page ix.

About Sun Management Center Software

Sun Management Center software is a convenient, single solution for managing multiple Sun systems, devices, and network resources. With its intuitive graphical interface based on Java™ software, Sun Management Center offers powerful management capabilities that allow you to do the following:

- Manage and monitor your server remotely from any location in the network
- Display physical and logical views of your exact server configuration
- Monitor system health conditions
- Access real-time system performance and configuration data, to diagnose potential capacity problems and performance bottlenecks
- Invoke SunVTS diagnostic software for online diagnosis of hardware problems
- Use predictive failure analysis features to warn of potential memory and disk hardware failures before they happen
- Organize systems by geographical location, server function, administrative responsibility, or other criteria for increased management flexibility
- Implement enterprise-wide security measures, such as authentication, data integrity, and access control lists

For More Information

Sun Management Center software is provided on a CD supplied in the Solaris media kit for your release. For information about installing and using Sun Management Center software, see the following documents provided with the Sun Management Center software:

- *Sun Management Center Software Installation Guide*
- *Sun Management Center Software User's Guide*
- *Sun Management Center Software Supplement for Workgroup Servers*

About Sun Remote System Control Software

Sun Remote System Control (RSC) software is a remote server management tool that allows you to monitor and control supported Sun servers over modem lines or over a network. The RSC software provides remote system administration for geographically distributed or physically inaccessible systems.

The RSC software works with the system controller card included in all Sun Fire V890 servers. The system controller card runs independently of the host server, and operates off of 5-volt standby power from the system's power supplies. This allows the system controller card to serve as a "lights out" management tool that continues to function even when the server operating system goes offline or the system is powered off.

The system controller card plugs in to a dedicated slot on the system I/O board and includes integrated serial and Ethernet interfaces. The card provides two ports that are accessible through an opening in the system rear panel:

- 10-Mbps Ethernet port via an RJ-45 twisted-pair Ethernet (TPE) connector
- EIA-232D serial port via an RJ-45 connector

Once RSC software is configured to manage your server, you can use it to run diagnostic tests, view diagnostic and error messages, reboot your server, and display environmental status information on a remote console. If the operating system is down, RSC can automatically notify you of any power failures, hardware failures, or other important events that may be occurring on your server.

RSC Capabilities

RSC software provides the following system administration capabilities:

- Access Solaris and OpenBoot PROM console functions remotely via the Ethernet port on the system controller card
- Run power-on self-test (POST) and OpenBoot Diagnostics from a remote console
- Remotely monitor server environmental conditions, such as fan, temperature, and power supply status, even when the server is offline
- View a graphical representation of the server's front panel, including keyswitch position and LED states
- Receive notification of server problems via email or pager, even in the event of a power failure

- Perform remote server reboot, power-on, and power-off functions on demand
- Access a detailed log of RSC events, command history, and detected errors

RSC complements existing Sun monitoring and diagnostic tools such as Sun Management Center, SunVTS, POST, and OpenBoot Diagnostics.

RSC User Interfaces

RSC offers the following user interfaces:

- A graphical user interface (GUI) that runs as a Java client application on workstations connected to the server through the RSC Ethernet interface or through a standard modem connection using Point-to-Point Protocol (PPP)
- A command-line interface (CLI) that you can access through the RSC Ethernet network, through a standard modem connection, or through an alphanumeric terminal attached directly to the RSC serial port

The GUI client application, based on Java software, runs on workstations using the the Solaris Operating System, Microsoft Windows 95, Windows 98, or Windows NT operating environments.

For More Information

Sun RSC software is included on the Supplement CD for your specific Solaris release. For installation instructions, see the *Solaris Sun Hardware Platform Guide* provided in the Solaris media kit. For information about configuring and using RSC, see the *Sun Remote System Control (RSC) User's Guide* provided with the RSC software.

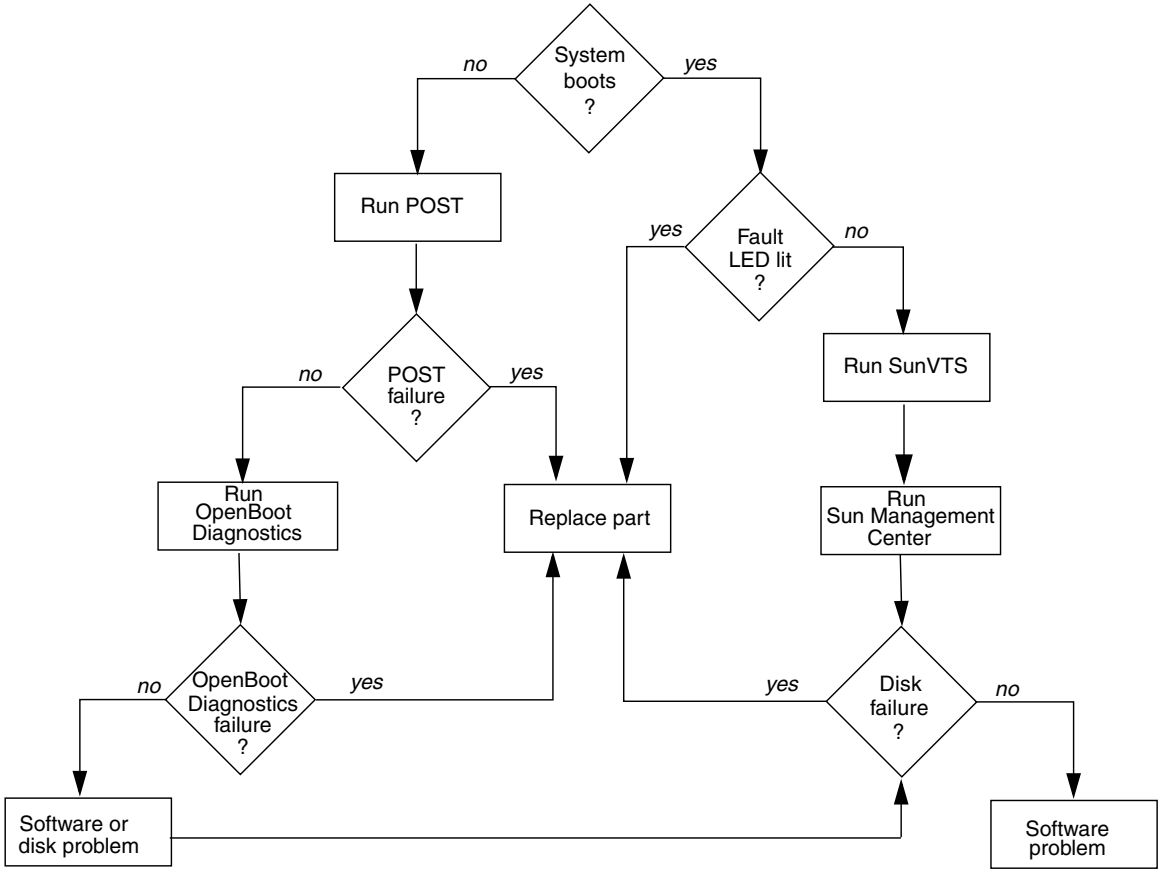
CHAPTER 2

Troubleshooting Strategies

This chapter provides information about error indications and software commands to help you determine which component you need to replace. It contains the following sections:

Note – The procedures in this chapter assume that you are familiar with the OpenBoot™ firmware and that you know how to enter the OpenBoot environment. For more information about the OpenBoot firmware, see the *OpenBoot 4.x Command Reference Manual*. An online version of the manual is included with the *OpenBoot Collection AnswerBook2* that ships with Solaris software.

The following chart shows which tools you can use to diagnose hardware and software problems.



About Troubleshooting Your System

The system provides the following features to help you identify and isolate hardware problems:

- Error indications
- Software commands
- Diagnostic tools

This section describes the error indications and software commands provided to help you troubleshoot your system. Diagnostic tools are covered in "About Diagnostic Tools" on page 2.

Error Indications

The system provides error indications via LEDs and error messages. Using the two in combination, you can isolate a problem to a particular field-replaceable uint (FRU) with a high degree of confidence.

The system provides status indicator LEDs in the following places:

- Front panel
- CPU/Memory board slots
- Power supplies
- Disk drives
- PCI slots
- Fan trays

Error messages are logged in the `/var/adm/messages` file and are also displayed on the system console by the diagnostic tools.

For additional information about LEDs, see the *Sun Fire V890 Server Owner's Guide*.

Status Indicator LEDs

Front panel LEDs provide your first indication that there is a problem with your system. Usually, a front panel LED is not the only indication of a problem. Error messages and other LEDs within the enclosure can help to isolate the problem further. For additional information about the front panel LEDs, see the *Sun Fire V890 Server Owner's Guide*.

The front panel LEDs provide general system status, alert you to system problems, and help you determine the location of system faults:

- At the top of the status and control panel, three general status LEDs provide a snapshot of the system status.
- Below the Power button and security keyswitch, a graphical display provides additional LED icons to indicate specific fault conditions and locations.

Located on the rear of each power supply, the power supply LEDs indicate:

- Whether the power supply has encountered a fault
- Whether the power input and outputs are functional and within acceptable limits

For additional information about the power supply LEDs, see the *Sun Fire V890 Server Owner's Guide*.

Fault LEDs within the enclosure help pinpoint the location of the faulty device. LEDs within the enclosure include:

- CPU/Memory board slot LEDs
- PCI slot LEDs
- CPU, I/O, and motherboard fan trays LEDs
- Disk drive LEDs

For detailed information about these LEDs, see the *Sun Fire V890 Server Owner's Guide*.

Since all front panel and power supply LEDs are powered by the system's 5-volt standby power source, fault LEDs remain illuminated for any fault condition that results in a system shutdown.

During system startup, the front panel LEDs are individually toggled on and off to verify that each one is working correctly.

Error Messages

Error messages and other system messages are saved in the file `/var/adm/messages`. The two firmware-based diagnostic tools, POST and OpenBoot Diagnostics, also display error messages in a standard format on the local system console or on an RSC console (if configured). See "Sample POST Error Messages" on page 14 and "Sample OpenBoot Diagnostics Error Messages" on page 37 for more information.

The amount of information displayed in OpenBoot Diagnostics messages is determined by the keywords specified for the OpenBoot configuration variable `test-args`. See "OpenBoot Configuration Variables for OpenBoot Diagnostics" on page 18 for additional details.

Software Commands

Several Solaris and OpenBoot firmware commands are available for diagnosing system problems. For more information about Solaris commands, see the appropriate man pages. For additional information about OpenBoot commands, see the *OpenBoot 4.x Command Reference Manual*. An online version of the manual is included with the *OpenBoot Collection AnswerBook* that ships with Solaris software.

Solaris `prtdiag` Command

The `prtdiag` command is a UNIX shell command used to display system configuration and diagnostic information. You can use the `prtdiag` command to display:

- System configuration, including information about clock frequencies, CPUs, memory, and I/O card types
- Diagnostic and environmental information
- Failed field-replaceable units (FRUs)

To run `prtdiag`, type:

```
% /usr/platform/sun4u/sbin/prtdiag
```

An example of `prtdiag` output follows.

```
% /usr/platform/sun4u/sbin/prtdiag
System Configuration:  Sun Microsystems  sun4u Sun Fire V890
System clock frequency: 150 MHz
Memory size: 32768 Megabytes

========================= CPUs ================================================

            Run   E$  CPU    CPU
Brd  CPU   MHz   MB Impl.   Mask
--- ----- ---- ---- ------- ----
 A  0, 16 1200 16.0 US-IV    2.4
 B  1, 17 1200 16.0 US-IV    2.4
 A  2, 18 1200 16.0 US-IV    2.4
 B  3, 19 1200 16.0 US-IV    2.4
 C  4, 20 1200 16.0 US-IV    2.4
 D  5, 21 1200 16.0 US-IV    2.4
 C  6, 22 1200 16.0 US-IV    2.4
 D  7, 23 1200 16.0 US-IV    2.4
```

```
========================= Memory Configuration ================================

           Logical  Logical  Logical
      MC   Bank     Bank     Bank         DIMM    Interleave  Interleaved
 Brd  ID   num      size     Status       Size    Factor      with
----  ---  ----     ------   -----------  ------  ----------  -----------
  A    0     0      1024MB   no_status     512MB     8-way        0
  A    0     1      1024MB   no_status     512MB     8-way        0
  A    0     2      1024MB   no_status     512MB     8-way        0
  A    0     3      1024MB   no_status     512MB     8-way        0
  B    1     0      1024MB   no_status     512MB     8-way        1
  B    1     1      1024MB   no_status     512MB     8-way        1
  B    1     2      1024MB   no_status     512MB     8-way        1
  B    1     3      1024MB   no_status     512MB     8-way        1
  A    2     0      1024MB   no_status     512MB     8-way        0
  A    2     1      1024MB   no_status     512MB     8-way        0
  A    2     2      1024MB   no_status     512MB     8-way        0
  A    2     3      1024MB   no_status     512MB     8-way        0
  B    3     0      1024MB   no_status     512MB     8-way        1
  B    3     1      1024MB   no_status     512MB     8-way        1
  B    3     2      1024MB   no_status     512MB     8-way        1
  B    3     3      1024MB   no_status     512MB     8-way        1
  C    4     0      1024MB   no_status     512MB     8-way        2
  C    4     1      1024MB   no_status     512MB     8-way        2
  C    4     2      1024MB   no_status     512MB     8-way        2
  C    4     3      1024MB   no_status     512MB     8-way        2
  D    5     0      1024MB   no_status     512MB     8-way        3
  D    5     1      1024MB   no_status     512MB     8-way        3
  D    5     2      1024MB   no_status     512MB     8-way        3
  D    5     3      1024MB   no_status     512MB     8-way        3
  C    6     0      1024MB   no_status     512MB     8-way        2
  C    6     1      1024MB   no_status     512MB     8-way        2
  C    6     2      1024MB   no_status     512MB     8-way        2
  C    6     3      1024MB   no_status     512MB     8-way        2
  D    7     0      1024MB   no_status     512MB     8-way        3
  D    7     1      1024MB   no_status     512MB     8-way        3
  D    7     2      1024MB   no_status     512MB     8-way        3
  D    7     3      1024MB   no_status     512MB     8-way        3


========================= IO Cards =========================
                          Bus  Max
     IO   Port Bus        Freq Bus  Dev,
Brd  Type  ID  Side Slot  MHz  Freq Func State Name                          Model
---- ----  ----  ---- ----  ---- ---- ---- ----- ----------------------------- ---------
I/O  PCI   8     B    2     33   33   3,0  ok    pci108e,1000-pci108e,1000.1
I/O  PCI   8     B    2     33   33   3,1  ok    SUNW,hme-pci108e,1001
I/O  PCI   8     B    0     33   33   5,0  ok    pci12de,200-pci12de,200.0
I/O  PCI   9     B    6     33   33   2,0  ok    pci108e,3de7-pci108e,3de7.0
I/O  PCI   9     A    8     33   66   2,0  ok    pci108e,3de7-pci108e,3de7.0
```

To isolate an intermittent failure, it may be helpful to maintain a `prtdiag` history log. Use `prtdiag` with the `-l` (log) option to send output to a log file in `/var/adm`.

To display environmental information, use `prtdiag` with the `-v` option. Type:

```
% /usr/platform/sun4u/sbin/prtdiag -v
```

The `prtdiag` command with the `-v` option produces all of the output of the `prtdiag` command (shown in the preceding example) in addition to environmental information, current keyswitch position, LED indications, and other information.

The following is an example of the additional output produced by the `-v` option.

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

System Temperatures (Celsius):
------------------------------
Device          Temperature     Status
---------------------------------------
CPU0             66             OK
CPU1             64             OK
CPU2             66             OK
CPU3             70             OK
CPU4             70             OK
CPU5             72             OK
CPU6             74             OK
CPU7             71             OK
MB               14             OK
IOB               3             OK
DBP0              5             OK

=================================

Front Status Panel:
-------------------
Keyswitch position: NORMAL

System LED Status:
                   GEN FAULT                REMOVE
                    [OFF]                    [OFF]

                   DISK FAULT               POWER FAULT
                    [OFF]                    [OFF]

                   LEFT THERMAL FAULT       RIGHT THERMAL FAULT
                    [OFF]                    [OFF]

                   LEFT DOOR                RIGHT DOOR
                    [OFF]                    [OFF]

=================================
```

```
Disk Status:
          Presence      Fault LED       Remove LED
DISK   0: [PRESENT]        [OFF]           [OFF]
DISK   1: [PRESENT]        [OFF]           [OFF]
DISK   2: [PRESENT]        [OFF]           [OFF]
DISK   3: [PRESENT]        [OFF]           [OFF]
DISK   4: [PRESENT]        [OFF]           [OFF]
DISK   5: [PRESENT]        [OFF]           [OFF]
DISK   6: [  EMPTY]
DISK   7: [  EMPTY]
DISK   8: [  EMPTY]
DISK   9: [  EMPTY]
DISK  10: [  EMPTY]
DISK  11: [  EMPTY]
==================================

Fan Bank :
----------

Bank                         Speed         Status        Fan State
                             ( RPMS )
----                        --------      ---------      ---------
CPU0_PRIM_FAN                 2127        [ENABLED]          OK
CPU1_PRIM_FAN                 2222        [ENABLED]          OK
CPU0_SEC_FAN                     0        [DISABLED]         OK
CPU1_SEC_FAN                     0        [DISABLED]         OK
IO0_PRIM_FAN                  3000        [ENABLED]          OK
IO1_PRIM_FAN                  2857        [ENABLED]          OK
IO0_SEC_FAN                      0        [DISABLED]         OK
IO1_SEC_FAN                      0        [DISABLED]         OK
IO_BRIDGE_PRIM_FAN            3614        [ENABLED]          OK
IO_BRIDGE_SEC_FAN                0        [DISABLED]         OK

==================================

Power Supplies:
---------------

Supply     Status     Fan Fail  Temp Fail  CS Fail  3.3V   5V   12V   48V
------  ------------  --------  ---------  -------  ----   --   ---   ---
PS0      GOOD                                         6     6     4     2
PS1      GOOD                                         7     6     4     2
PS2      GOOD                                         7     6     4     2


========================= HW Revisions =======================================

System PROM revisions:
----------------------
OBP 4.15.1 2004/06/02 16:06

IO ASIC revisions:
------------------
         Port
Model     ID  Status Version
--------  ----  ------ -------
Schizo    8     ok      7
Schizo    9     ok      7
```

Note – Refer to the `prtdiag` man page for additional information.

Solaris `prtconf` Command

The `prtconf` command displays system configuration information, including the total amount of memory and the device configuration as described by the system's device hierarchy.

To run `prtconf`, type:

```
% /usr/sbin/prtconf
```

The following is partial sample output.

```
% /usr/sbin/prtconf
System Configuration:  Sun Microsystems  sun4u
Memory size: 3072 Megabytes
System Peripherals (Software Nodes):

SUNW,Sun-Fire-V890
    packages (driver not attached)
        SUNW,builtin-drivers (driver not attached)
        deblocker (driver not attached)
        disk-label (driver not attached)
        terminal-emulator (driver not attached)
        obp-tftp (driver not attached)
        SUNW,debug (driver not attached)
        dropins (driver not attached)
        kbd-translator (driver not attached)
        ufs-file-system (driver not attached)
    chosen (driver not attached)
    openprom (driver not attached)
        client-services (driver not attached)
    options, instance #0
    aliases (driver not attached)
    memory (driver not attached)
    virtual-memory (driver not attached)
```

```
    cmp (driver not attached)
        cpu (driver not attached)
        cpu (driver not attached)
    memory-controller, instance #0
    cmp (driver not attached)
        cpu (driver not attached)
        cpu (driver not attached)
    memory-controller, instance #1
    cmp (driver not attached)
        cpu (driver not attached)
        cpu (driver not attached)
    memory-controller, instance #2
    cmp (driver not attached)
        cpu (driver not attached)
        cpu (driver not attached)
    memory-controller, instance #3
    cmp (driver not attached)
        cpu (driver not attached)
        cpu (driver not attached)
    memory-controller, instance #4
    cmp (driver not attached)
        cpu (driver not attached)
        cpu (driver not attached)
    memory-controller, instance #5
    cmp (driver not attached)
        cpu (driver not attached)
        cpu (driver not attached)
    memory-controller, instance #6
    cmp (driver not attached)
        cpu (driver not attached)
        cpu (driver not attached)
```

```
pci, instance #0
        scsi, instance #0
            disk (driver not attached)
            tape (driver not attached)
            sd, instance #0 (driver not attached)
            sd, instance #1 (driver not attached)
            sd, instance #2 (driver not attached)
TSI,gfxp (driver not attached)
    pci, instance #1
        network (driver not attached)
        SUNW,qlc, instance #0
            fp (driver not attached)
                disk (driver not attached)
            fp, instance #0
                ses (driver not attached)
                ssd, instance #0
                ssd, instance #1
                ssd, instance #2
                ssd, instance #3
pci, instance #2
        ebus, instance #0
            flashprom (driver not attached)
bbc, instance #0
            power (driver not attached)
            i2c, instance #1
                fru, instance #0
                fru, instance #1
                fru, instance #2
                fru, instance #3
                fru, instance #4
                fru, instance #5
                fru, instance #6
                fru, instance #7
                temperature, instance #0
                temperature, instance #1
                temperature, instance #2
                temperature, instance #3
                temperature, instance #4
                temperature, instance #5
                temperature, instance #6
```

Solaris `prtfru` Command

The `prtfru` command displays specific information about the following FRUs:

- I/O board
- System controller card
- Power distribution board and power supplies
- FC-AL backplane
- Motherboard
- CPU/Memory boards
- DIMMs

The `prtfru` command also displays the contents of the FRU SEEPROMs:

- FRU description
- Part number and serial number
- Hardware revision levels
- Temperature, voltage, and power data

The following is partial sample output from the `prtfru` command.

```
% prtfru
/frutree
/frutree/chassis (fru)
/frutree/chassis/io-board (container)
   SEGMENT: SD
      /ManR
      /ManR/UNIX_Timestamp32: Tue May  9 09:36:08 EDT 2000
      /ManR/Fru_Description: ASSY,PCB,PCI/IO,BRD,RHINO
      /ManR/Manufacture_Loc: BENCHMARK ELECTRONICS INC,HUNTSVILLE,ALABAMA,USA
      /ManR/Sun_Part_No: 5015142
      /ManR/Sun_Serial_No: 000069
      /ManR/Vendor_Name: NO JEDEC CODE FOR THIS VENDOR
      /ManR/Initial_HW_Dash_Level: 03
      /ManR/Initial_HW_Rev_Level: 01
      /ManR/Fru_Shortname: /Dak_IOBoardR
      /Dak_IOBoardR/PROM_Format_Version: 1
      /Dak_IOBoardR/Ambient_Temp_Array: 9223372036854775806
      /Dak_IOBoardR/Min_Power_Rating (4 iterations)
      /Dak_IOBoardR/Min_Power_Rating[0]: 11
      /Dak_IOBoardR/Min_Power_Rating[1]: 22
      /Dak_IOBoardR/Min_Power_Rating[2]: 33
      /Dak_IOBoardR/Min_Power_Rating[3]: 44
```

```
 /Dak_IOBoardR/Max_Power_Rating (4 iterations)
      /Dak_IOBoardR/Max_Power_Rating[0]: 22
      /Dak_IOBoardR/Max_Power_Rating[1]: 33
      /Dak_IOBoardR/Max_Power_Rating[2]: 44
      /Dak_IOBoardR/Max_Power_Rating[3]: 55
/frutree/chassis/rsc-board (container)
   SEGMENT: SD
/frutree/chassis/fcal-backplane-slot?Label=0
/frutree/chassis/fcal-backplane-slot?Label=0/fcal-backplane (container)
/frutree/chassis/fcal-backplane-slot?Label=1
/frutree/chassis/fcal-backplane-slot?Label=1/fcal-backplane (container)
/frutree/chassis/power-dist-board (container)
/frutree/chassis/power-dist-board/power-supply-slot?Label=0
/frutree/chassis/power-dist-board/power-supply-slot?Label=0/power-supply (container)
/frutree/chassis/power-dist-board/power-supply-slot?Label=1
/frutree/chassis/power-dist-board/power-supply-slot?Label=1/power-supply (container)
/frutree/chassis/power-dist-board/power-supply-slot?Label=2
/frutree/chassis/power-dist-board/power-supply-slot?Label=2/power-supply (container)
/frutree/chassis/system-board (container)
/frutree/chassis/system-board/cpu-mem-slot?Label=A
/frutree/chassis/system-board/cpu-mem-slot?Label=B
/frutree/chassis/system-board/cpu-mem-slot?Label=B/cpu-mem-module (container)
SEGMENT: SD
      /ManR
      /ManR/UNIX_Timestamp32: Mon Jun 12 14:31:06 EDT 2000
      /ManR/Fru_Description: ASSY,CPU,DUAL,DAK
      /ManR/Manufacture_Loc: BENCHMARK ELECTRONICS INC, HUNTSVILLE, AL, USA
      /ManR/Sun_Part_No: 5014150
      /ManR/Sun_Serial_No: 001135
      /ManR/Vendor_Name: NO JEDEC CODE FOR THIS VENDOR
      /ManR/Initial_HW_Dash_Level: 03
      /ManR/Initial_HW_Rev_Level: 06
      /ManR/Fru_Shortname:
```

Solaris `prtpicl` Command

The `prtpicl` command displays the name and Platform Information and Control Library (PICL) class of all nodes in the PICL tree.

To display the high temperature and low temperature critical thresholds for each component, use the `prtpicl -v` option. See "Environmental Failures" on page 86 for more information.

The following is partial sample output from the `prtpicl` command.

```
% prtpicl
  / (picl, 4300000001)
     SYSTEM (picl, 4300000005)
         MOTHERBOARD (picl, 430000000a)
             CPU0_PFAN_TACH (fan-tachometer, 43000000e5)
             CPU1_PFAN_TACH (fan-tachometer, 43000000ef)
             CPU0_SFAN_TACH (fan-tachometer, 43000000f9)
             CPU1_SFAN_TACH (fan-tachometer, 4300000103)
             IO_BRIDGE_PFAN_TACH (fan-tachometer, 4300000135)
             IO_BRIDGE_SFAN_TACH (fan-tachometer, 430000013f)
             IO_PFAN_ONOFF_SWITCH (switch, 430000015a)
             IO_SFAN_ONOFF_SWITCH (switch, 430000015f)
             IO_BRIDGE_PFAN_ONOFF_SWITCH (switch, 4300000164)
             IO_BRIDGE_SFAN_ONOFF_SWITCH (switch, 4300000169)
             DISK_BP0_PR_SENSOR (gpio, 430000016e)
             DISK_BP1_PR_SENSOR (gpio, 4300000175)
             RSC_PR_SENSOR (gpio, 430000017c)
             CPU_0_2_MOD_SLOT (picl, 4300000274)
             CPU_1_3_MOD_SLOT (picl, 4300000279)
                 CPU_1_3_MOD_CARD (picl, 430000028d)
                   CPU1_DIE_TEMPERATURE_SENSOR (temperature-sensor, 43000002f2)
                   CPU3_DIE_TEMPERATURE_SENSOR (temperature-sensor, 4300000306)
                      24C64_A0_1 (i2c, 4300000696)
                      24C64_A2_1 (i2c, 430000069b)
                      24C64_A4_1 (i2c, 43000006a0)
                      24C64_A6_1 (i2c, 43000006a5)
                      24C64_A8_1 (i2c, 43000006aa)
                      24C64_AA_1 (i2c, 43000006af)
                      24C64_AC_1 (i2c, 43000006b4)
                      24C64_AE_1 (i2c, 43000006b9)
                      24C64_A0_3 (i2c, 43000006e6)
                      24C64_A2_3 (i2c, 43000006eb)
                      24C64_A4_3 (i2c, 43000006f0)
                      24C64_A6_3 (i2c, 43000006f5)
                      24C64_A8_3 (i2c, 43000006fa)
```

```
CPU_4_6_MOD_SLOT (picl, 430000027e)
            CPU_5_7_MOD_SLOT (picl, 4300000283)
            CPU_0_2_MOD_PR_SENSOR (gpio, 43000002cc)
            CPU_1_3_MOD_PR_SENSOR (gpio, 43000002d3)
            CPU_4_6_MOD_PR_SENSOR (gpio, 43000002da)
            CPU_5_7_MOD_PR_SENSOR (gpio, 43000002e1)
            DAR8_DIE_TEMPERATURE_SENSOR (temperature-sensor, 430000034c)
            DCS8_DIE_TEMPERATURE_SENSOR (temperature-sensor, 4300000356)
            24C64_A8_4 (i2c, 4300000718)
            SSC050_80_5 (i2c, 4300000786)
            HPC3130_EC_5 (i2c, 43000007fe)
            24C64_A0_11 (i2c, 43000008bc)
        IO_BOARD (picl, 430000000f)
            FAN_BLAST_OFF_SWITCH (switch, 430000003c)
            CPU_PFAN_PR_SENSOR (gpio, 43000000bb)
            CPU_SFAN_PR_SENSOR (gpio, 43000000c2)
            IO_PFAN_PR_SENSOR (gpio, 43000000c9)
            IO_SFAN_PR_SENSOR (gpio, 43000000d0)
            IO_BRIDGE_PFAN_PR_SENSOR (gpio, 43000000d7)
            IO_BRIDGE_SFAN_PR_SENSOR (gpio, 43000000de)
            IO0_PFAN_TACH (fan-tachometer, 430000010d)
            IO1_PFAN_TACH (fan-tachometer, 4300000117)
            IO0_SFAN_TACH (fan-tachometer, 4300000121)
           PS1_PR_SENSOR (gpio, 430000022a)
           PS1_PR_SENSOR (gpio, 430000022a)
           PS2_PR_SENSOR (gpio, 430000026d)
            MB_AMB_TEMPERATURE_SENSOR (temperature-sensor, 4300000338)
            IOB_AMB_TEMPERATURE_SENSOR (temperature-sensor, 4300000342)
            PCI0_SLOT (picl, 43000003b4)
                PCI0_CARD (picl, 43000003e1)
            PCI1_SLOT (picl, 43000003b9)
            PCI2_SLOT (picl, 43000003be)
            PCI3_SLOT (picl, 43000003c3)
            PCI4_SLOT (picl, 43000003c8)
            PCI5_SLOT (picl, 43000003cd)
            PCI6_SLOT (picl, 43000003d2)
            PCI7_SLOT (picl, 43000003d7)
            PCI8_SLOT (picl, 43000003dc)
            PCI0_PR_SENSOR (gpio, 430000047a)
           PCI1_PR_SENSOR (gpio, 4300000481)
            PCI2_PR_SENSOR (gpio, 4300000488)
            PCI3_PR_SENSOR (gpio, 430000048f)
            PCI4_PR_SENSOR (gpio, 4300000496)
            PCI5_PR_SENSOR (gpio, 430000049d)
            PCI6_PR_SENSOR (gpio, 43000004a4)
```

Solaris `showrev` Command

The `showrev` command displays revision information for the current hardware and software. When used with the `-p` option, this command displays installed patches.

The following is partial sample output from the `showrev` command with the `-p` option.

```
% /usr/sbin/showrev -p
Patch: 109729-01 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWcsu
Patch: 109783-01 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWcsu
Patch: 109807-01 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWcsu
Patch: 109809-01 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWcsu
Patch: 110905-01 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWcsu
Patch: 110910-01 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWcsu
Patch: 110914-01 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWcsu
Patch: 108964-04 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWcsr
```

Solaris `psrinfo` Command

The `psrinfo` command displays the date and time each CPU came online.

The `psrinfo` command with the `-v` option displays additional information about the CPUs, including clock speed.

The following is sample output from the `psrinfo` command with the `-v` option.

```
% /usr/sbin/psrinfo -v
Status of processor 0 as of: 04/11/01 12:03:45
  Processor has been on-line since 04/11/01 10:53:03.
  The sparcv9 processor operates at 750 MHz,
        and has a sparcv9 floating point processor.
Status of processor 2 as of: 04/11/01 12:03:45
  Processor has been on-line since 04/11/01 10:53:05.
  The sparcv9 processor operates at 750 MHz,
        and has a sparcv9 floating point processor.
```

OpenBoot `show-devs` Command

If you are working from the `ok` prompt, you can use the OpenBoot `show-devs` command to list the devices in the system configuration. The following is sample `show-devs` output for a Sun Fire V890 server configured with a full complement of CPU/Memory boards, DIMMs, power supplies, and FC-AL disk backplanes. The system also includes a Sun StorEdge™ Dual Fibre Channel Host Adapter card to drive Loop B of the FC-AL mass storage subsystem. The `show-devs` output displays the device tree for the system. Helpful descriptions for most of the devices are provided to the right of the sample output.

```
ok show-devs
/pci@9,600000                                   PCI Bus C - Slots 7 and 8
/pci@9,700000                                   PCI Bus D - Slots 4, 5, 6, RIO ASIC
/pci@8,600000                                   PCI Bus A - FC-AL, Gigabit Ethernet
/pci@8,700000                                   PCI Bus B - Slots 0-3, SCSI controller
/memory-controller@7,400000                     Memory controller (CPU7) - Slot D
/cmp@7,0                                        CPU7 - Slot D
/memory-controller@6,400000                     Memory controller (CPU6) - Slot C
/cmp@6,0                                        CPU6 - Slot C
/memory-controller@5,400000                     Memory controller (CPU5) - Slot D
/cmp@5,0                                        CPU5 - Slot D
/memory-controller@4,400000                     Memory controller (CPU4) - Slot C
/cmp@4,0                                        CPU4 - Slot C
/memory-controller@3,400000                     Memory controller (CPU3) - Slot B
/cmp@3,0                                        CPU3 - Slot B
/memory-controller@2,400000                     Memory controller (CPU2) - Slot A
/cmp@2,0                                        CPU2 - Slot A
/memory-controller@1,400000                     Memory controller (CPU1) - Slot B
/cmp@1,0                                        CPU1 - Slot B
/memory-controller@0,400000                     Memory controller (CPU0) - Slot A
/cmp@0,0                                        CPU0 - Slot A
/virtual-memory
/memory@m0,20
/aliases
/options
/openprom
/chosen
/packages
/pci@9,600000/pci@1
/pci@9,600000/pci@1/SUNW,qlc@5                  ISP2200A PCI FC-AL controller, external
/pci@9,600000/pci@1/SUNW,qlc@4                  ISP2200A PCI FC-AL controller (Loop B)
/pci@9,600000/pci@1/SUNW,qlc@5/fp@0,0
/pci@9,600000/pci@1/SUNW,qlc@5/fp@0,0/disk
```

/pci@9,700000/ebus@1/serial@1,400000	Serial ports A and B
/pci@9,700000/ebus@1/rsc-console@1,3083f8	system controller card
/pci@9,700000/ebus@1/rsc-control@1,3062f8	system controller card
/pci@9,700000/ebus@1/pmc@1,300700	Power management controller
/pci@9,700000/ebus@1/gpio@1,300600	Super I/O subsystem
/pci@9,700000/ebus@1/rtc@1,300070	Real time clock
/pci@9,700000/ebus@1/i2c@1,500030	I^2C segment 11 (NVRAM)
/pci@9,700000/ebus@1/i2c@1,50002e	I^2C segments 6 - 10 (FRU PROMs)
/pci@9,700000/ebus@1/bbc@1,500000	Boot bus controller
/pci@9,700000/ebus@1/i2c@1,30	I^2C segment 5 (envrironmental)
/pci@9,700000/ebus@1/i2c@1,2e	I^2C segment 0 - 4 (FRU PROMs)
/pci@9,700000/ebus@1/power@1,30002e	
/pci@9,700000/ebus@1/bbc@1,0	Boot bus controller
/pci@9,700000/ebus@1/flashprom@0,0	OpenBoot PROM
/pci@9,700000/ebus@1/i2c@1,500030/idprom@0,a0	"NVRAM" SEEPROM
/pci@9,700000/ebus@1/i2c@1,500030/nvram@0,a0	
/pci@9,700000/ebus@1/i2c@1,50002e/temperature@4,56	MAX1617 die thermal sensor, MDR8-5
/pci@9,700000/ebus@1/i2c@1,50002e/temperature@4,54	MAX1617 die thermal sensor, MDR8-4
/pci@9,700000/ebus@1/i2c@1,50002e/temperature@4,52	MAX1617 die thermal sensor, MDR8-3
/pci@9,700000/ebus@1/i2c@1,50002e/fru@4,a2	FRU SEEPROM – CPU/Memory board D
/pci@9,700000/ebus@1/i2c@1,50002e/fru@4,a0	FRU SEEPROM – CPU/Memory board C
/pci@9,700000/ebus@1/i2c@1,50002e/fru@3,ae	FRU SEEPROM – CPU7 DIMM J8201
/pci@9,700000/ebus@1/i2c@1,50002e/fru@3,ac	FRU SEEPROM – CPU7 DIMM J8001
/pci@9,700000/ebus@1/i2c@1,50002e/fru@3,aa	FRU SEEPROM – CPU7 DIMM J8200
/pci@9,700000/ebus@1/i2c@1,50002e/fru@3,a8	FRU SEEPROM – CPU7 DIMM J8000
/pci@9,700000/ebus@1/i2c@1,50002e/fru@3,a6	FRU SEEPROM – CPU7 DIMM J8101
/pci@9,700000/ebus@1/i2c@1,50002e/fru@3,a4	FRU SEEPROM – CPU7 DIMM J7901
/pci@9,700000/ebus@1/i2c@1,50002e/fru@3,a2	FRU SEEPROM – CPU7 DIMM J8100
/pci@9,700000/ebus@1/i2c@1,50002e/fru@3,a0	FRU SEEPROM – CPU7 DIMM J7900
/pci@9,700000/ebus@1/i2c@1,50002e/fru@2,ae	FRU SEEPROM – CPU6 DIMM J8201
/pci@9,700000/ebus@1/i2c@1,50002e/fru@2,ac	FRU SEEPROM – CPU6 DIMM J8001
/pci@9,700000/ebus@1/i2c@1,50002e/fru@2,aa	FRU SEEPROM – CPU6 DIMM J8200
/pci@9,700000/ebus@1/i2c@1,50002e/fru@2,a8	FRU SEEPROM – CPU6 DIMM J8000
/pci@9,700000/ebus@1/i2c@1,50002e/fru@2,a6	FRU SEEPROM – CPU6 DIMM J8101
/pci@9,700000/ebus@1/i2c@1,50002e/fru@2,a4	FRU SEEPROM – CPU6 DIMM J7901
/pci@9,700000/ebus@1/i2c@1,50002e/fru@2,a2	FRU SEEPROM – CPU6 DIMM J8100
/pci@9,700000/ebus@1/i2c@1,50002e/fru@2,a0	FRU SEEPROM – CPU6 DIMM J7900
/pci@9,700000/ebus@1/i2c@1,50002e/fru@1,ae	FRU SEEPROM – CPU5 DIMM J3201
/pci@9,700000/ebus@1/i2c@1,50002e/fru@1,ac	FRU SEEPROM – CPU5 DIMM J3001
/pci@9,700000/ebus@1/i2c@1,50002e/fru@1,aa	FRU SEEPROM – CPU5 DIMM J3200
/pci@9,700000/ebus@1/i2c@1,50002e/fru@1,a8	FRU SEEPROM – CPU5 DIMM J3000
/pci@9,700000/ebus@1/i2c@1,50002e/fru@1,a6	FRU SEEPROM – CPU5 DIMM J3101
/pci@9,700000/ebus@1/i2c@1,50002e/fru@1,a4	FRU SEEPROM – CPU5 DIMM J2901
/pci@9,700000/ebus@1/i2c@1,50002e/fru@1,a2	FRU SEEPROM – CPU5 DIMM J3100

```
/pci@9,700000/ebus@1/i2c@1,30/hotplug-controller@0,ecHot plug controller, CPU/Memory slots
/pci@9,700000/ebus@1/i2c@1,30/hotplug-controller@0,e8Hot plug controller, PCI slots 5-8
/pci@9,700000/ebus@1/i2c@1,30/hotplug-controller@0,e6Hot plug controller, PCI slots 2-8
/pci@9,700000/ebus@1/i2c@1,30/hotplug-controller@0,e2Hot plug controller, PCI slots 0-1
/pci@9,700000/ebus@1/i2c@1,30/rscrtc@0,d0            system controller card real time clock
/pci@9,700000/ebus@1/i2c@1,30/fru@0,ae               FRU SEEPROM - power distribution board
/pci@9,700000/ebus@1/i2c@1,30/fru@0,ac               FRU SEEPROM - expansion backplane
/pci@9,700000/ebus@1/i2c@1,30/fru@0,a8               FRU SEEPROM - base backplane
/pci@9,700000/ebus@1/i2c@1,30/fru@0,a6               FRU SEEPROM - system controller card
/pci@9,700000/ebus@1/i2c@1,30/fru@0,a4               FRU SEEPROM - power supply 2
/pci@9,700000/ebus@1/i2c@1,30/fru@0,a2               FRU SEEPROM - power supply 1
/pci@9,700000/ebus@1/i2c@1,30/fru@0,a0               FRU SEEPROM - power supply 0
/pci@9,700000/ebus@1/i2c@1,30/temperature-sensor@0,9eLM75 thermal sensor, exp backplane
/pci@9,700000/ebus@1/i2c@1,30/temperature-sensor@0,9cLM75 thermal sensor, base backplane
/pci@9,700000/ebus@1/i2c@1,30/temperature@0,9a       MAX1617 die thermal sensor - CPU7
/pci@9,700000/ebus@1/i2c@1,30/temperature@0,98       MAX1617 die thermal sensor - CPU6
/pci@9,700000/ebus@1/i2c@1,30/adio@0,96              I/O board ambient temperature ADC
/pci@9,700000/ebus@1/i2c@1,30/adio@0,94              Current output monitor - power supply 2
/pci@9,700000/ebus@1/i2c@1,30/adio@0,92              Current output monitor - power supply 1
/pci@9,700000/ebus@1/i2c@1,30/adio@0,90              Current output monitor - power supply 0
/pci@9,700000/ebus@1/i2c@1,30/ioexp@0,8e             SSC-050 - expansion backplane
/pci@9,700000/ebus@1/i2c@1,30/ioexp@0,8c             SSC-050 - expansion backplane
/pci@9,700000/ebus@1/i2c@1,30/ioexp@0,8a             SSC-050 - base backplane
/pci@9,700000/ebus@1/i2c@1,30/ioexp@0,88             SSC-050 - base backplane
/pci@9,700000/ebus@1/i2c@1,30/ioexp@0,82             SSC-050 - I/O board
/pci@9,700000/ebus@1/i2c@1,30/ioexp@0,80             SSC-050 - motherboard
/pci@9,700000/ebus@1/i2c@1,30/ioexp@0,74
/pci@9,700000/ebus@1/i2c@1,30/ioexp@0,72
/pci@9,700000/ebus@1/i2c@1,30/ioexp@0,70
/pci@9,700000/ebus@1/i2c@1,30/i2c-bridge@0,60
/pci@9,700000/ebus@1/i2c@1,30/adio@0,5e              I/O board CPU fan speed control DAC
/pci@9,700000/ebus@1/i2c@1,30/controller@0,5c
/pci@9,700000/ebus@1/i2c@1,30/adio@0,5a
/pci@9,700000/ebus@1/i2c@1,30/controller@0,58
/pci@9,700000/ebus@1/i2c@1,30/temperature@0,56       MAX1617 die thermal sensor - CPU5
/pci@9,700000/ebus@1/i2c@1,30/temperature@0,54       MAX1617 die thermal sensor - CPU4
/pci@9,700000/ebus@1/i2c@1,30/temperature@0,52       MAX1617 die thermal sensor - CPU3
/pci@9,700000/ebus@1/i2c@1,30/ioexp@0,46             I/O board fan tray OK-to-Remove LEDs
/pci@9,700000/ebus@1/i2c@1,30/temperature@0,34       MAX1617 die thermal sensor - CPU2
/pci@9,700000/ebus@1/i2c@1,30/temperature@0,32       MAX1617 die thermal sensor - CPU1
/pci@9,700000/ebus@1/i2c@1,30/temperature@0,30       MAX1617 die thermal sensor - CPU0
/pci@9,700000/ebus@1/i2c@1,30/controller@0,1e        SSC100 controller Loop B exp backplane
/pci@9,700000/ebus@1/i2c@1,30/controller@0,1c        SSC100 controller Loop A exp backplane
/pci@9,700000/ebus@1/i2c@1,30/controller@0,1a        SSC100 controller Loop B base backplane
```

/pci@9,700000/ebus@1/i2c@1,2e/fru@4,aa	FRU SEEPROM – I/O board
/pci@9,700000/ebus@1/i2c@1,2e/fru@4,a8	FRU SEEPROM – motherboard
/pci@9,700000/ebus@1/i2c@1,2e/fru@4,a2	FRU SEEPROM – CPU/Memory board B
/pci@9,700000/ebus@1/i2c@1,2e/fru@4,a0	FRU SEEPROM – CPU/Memory board A
/pci@9,700000/ebus@1/i2c@1,2e/fru@3,ae	FRU SEEPROM – CPU3 DIMM J8201
/pci@9,700000/ebus@1/i2c@1,2e/fru@3,ac	FRU SEEPROM – CPU3 DIMM J8001
/pci@9,700000/ebus@1/i2c@1,2e/fru@3,aa	FRU SEEPROM – CPU3 DIMM J8200
/pci@9,700000/ebus@1/i2c@1,2e/fru@3,a8	FRU SEEPROM – CPU3 DIMM J8000
/pci@9,700000/ebus@1/i2c@1,2e/fru@3,a6	FRU SEEPROM – CPU3 DIMM J8101
/pci@9,700000/ebus@1/i2c@1,2e/fru@3,a4	FRU SEEPROM – CPU3 DIMM J7901
/pci@9,700000/ebus@1/i2c@1,2e/fru@3,a2	FRU SEEPROM – CPU3 DIMM J8100
/pci@9,700000/ebus@1/i2c@1,2e/fru@3,a0	FRU SEEPROM – CPU3 DIMM J7900
/pci@9,700000/ebus@1/i2c@1,2e/fru@2,ae	FRU SEEPROM – CPU2 DIMM J8201
/pci@9,700000/ebus@1/i2c@1,2e/fru@2,ac	FRU SEEPROM – CPU2 DIMM J8001
/pci@9,700000/ebus@1/i2c@1,2e/fru@2,aa	FRU SEEPROM – CPU2 DIMM J8200
/pci@9,700000/ebus@1/i2c@1,2e/fru@2,a8	FRU SEEPROM – CPU2 DIMM J8000
/pci@9,700000/ebus@1/i2c@1,2e/fru@2,a6	FRU SEEPROM – CPU2 DIMM J8101
/pci@9,700000/ebus@1/i2c@1,2e/fru@2,a4	FRU SEEPROM – CPU2 DIMM J7901
/pci@9,700000/ebus@1/i2c@1,2e/fru@2,a2	FRU SEEPROM – CPU2 DIMM J8100
/pci@9,700000/ebus@1/i2c@1,2e/fru@2,a0	FRU SEEPROM – CPU2 DIMM J7900
/pci@9,700000/ebus@1/i2c@1,2e/fru@1,ae	FRU SEEPROM – CPU1 DIMM J3201
/pci@9,700000/ebus@1/i2c@1,2e/fru@1,ac	FRU SEEPROM – CPU1 DIMM J3001
/pci@9,700000/ebus@1/i2c@1,2e/fru@1,aa	FRU SEEPROM – CPU1 DIMM J3200
/pci@9,700000/ebus@1/i2c@1,2e/fru@1,a8	FRU SEEPROM – CPU1 DIMM J3000
/pci@9,700000/ebus@1/i2c@1,2e/fru@1,a6	FRU SEEPROM – CPU1 DIMM J3101
/pci@9,700000/ebus@1/i2c@1,2e/fru@1,a4	FRU SEEPROM – CPU1 DIMM J2901
/pci@9,700000/ebus@1/i2c@1,2e/fru@1,a2	FRU SEEPROM – CPU1 DIMM J3100
/pci@9,700000/ebus@1/i2c@1,2e/fru@1,a0	FRU SEEPROM – CPU1 DIMM J2900
/pci@9,700000/ebus@1/i2c@1,2e/fru@0,ae	FRU SEEPROM – CPU0 DIMM J3201
/pci@9,700000/ebus@1/i2c@1,2e/fru@0,ac	FRU SEEPROM – CPU0 DIMM J3001
/pci@9,700000/ebus@1/i2c@1,2e/fru@0,aa	FRU SEEPROM – CPU0 DIMM J3200
/pci@9,700000/ebus@1/i2c@1,2e/fru@0,a8	FRU SEEPROM – CPU0 DIMM J3000
/pci@9,700000/ebus@1/i2c@1,2e/fru@0,a6	FRU SEEPROM – CPU0 DIMM J3101
/pci@9,700000/ebus@1/i2c@1,2e/fru@0,a4	FRU SEEPROM – CPU0 DIMM J2901
/pci@9,700000/ebus@1/i2c@1,2e/fru@0,a2	FRU SEEPROM – CPU0 DIMM J3100
/pci@9,700000/ebus@1/i2c@1,2e/fru@0,a0	FRU SEEPROM – CPU0 DIMM J2900
/pci@8,600000/SUNW,qlc@2	On-board FC-AL controller
/pci@8,600000/network@1	On-board Gigabit Ethernet interface
/pci@8,600000/SUNW,qlc@2/fp@0,0	
/pci@8,600000/SUNW,qlc@2/fp@0,0/disk	
/packages/disk-label	

```
/pci@8,700000/ide@1
/pci@8,700000/ide@1/cdrom
/pci@8,700000/ide@1/disk
/openprom/client-services
/packages/kbd-translator
/packages/dropins
/packages/SUNW,debug
/packages/obp-tftp
/packages/terminal-emulator
/packages/disk-label
/packages/deblocker
/packages/SUNW,builtin-drivers
```

OpenBoot `.env` Command

Use the OpenBoot `.env` command to display the current environmental status information.

The following is sample output from the `.env` command.

```
ok .env
Environmental Status:

Power Supplies:
PS0:                      Present, receiving AC power
PS1:                      Present, receiving AC power
PS2:                      Present, receiving AC power

Fans:
Tray 1 (CPU):           Present, Fan A @ 3225 RPM, Fan B @ 3191 RPM
Tray 2 (CPU):           Present, Fan A @ 3614 RPM, Fan B @ 3571 RPM
Tray 3 (I/O):           Present, Fan A @ 3488 RPM, Fan B @ 3409 RPM
Tray 4 (I/O):           Present, Fan A @ 3157 RPM, Fan B @ 3061 RPM
Fan  5 (IO-Bridge):       Present, Fan   @ 3846 RPM
Fan  6 (IO-Bridge):       Present, Fan   @ 3529 RPM

Temperatures:
CMP0:                     Ambient =  40 deg. C, Die =  61 deg. C
CMP1:                     Ambient =  40 deg. C, Die =  57 deg. C
CMP2:                     Ambient =  39 deg. C, Die =  59 deg. C
CMP3:                     Ambient =  39 deg. C, Die =  61 deg. C
CMP4:                     Ambient =  45 deg. C, Die =  63 deg. C
CMP5:                     Ambient =  41 deg. C, Die =  58 deg. C
CMP6:                     Ambient =  42 deg. C, Die =  65 deg. C
CMP7:                     Ambient =  40 deg. C, Die =  61 deg. C
Motherboard:              Ambient =  29 deg. C
I/O Board:                Ambient =  22 deg. C
Disk Backplane 0:         Ambient =  21 deg. C

Environmental monitor is ON
```

OpenBoot `printenv` Command

Use the OpenBoot `printenv` command to display the OpenBoot configuration variables. The display includes the current values for these variables as well as the default values.

The following is sample output for the `printenv` command.

```
ok printenv
Variable Name         Value                          Default Value

test-args
diag-passes           1                               1
local-mac-address?    true                            false
scsi-initiator-id     7                               7
oem-logo                                              No default
oem-logo?             false                           false
oem-banner                                            No default
oem-banner?           false                           false
ansi-terminal?        true                            true
screen-#columns       80                              80
screen-#rows          34                              34
ttyb-rts-dtr-off      false                           false
ttyb-ignore-cd        true                            true
ttya-rts-dtr-off      false                           false
ttya-ignore-cd        true                            true
ttyb-mode             9600,8,n,1,-                    9600,8,n,1,-
ttya-mode             9600,8,n,1,-                    9600,8,n,1,-
output-device         ttya                            screen
input-device          ttya                            keyboard
auto-boot-on-error?   true                            true
load-base             16384                           16384
auto-boot?            false                           true
boot-command          boot                            boot
diag-file
diag-device           disk net                        net
boot-file
boot-device           /pci@8,600000/SUNW,qlc@2 ...    disk net
use-nvramrc?          false                           false
nvramrc
security-mode         none                            No default
security-password                                     No default
security-#badlogins   0                               No default
verbosity             debug                           normal
fcode-debug?          false                           false
diag-out-console      false                           false
diag-trigger          none                            error-reset
power-on-res ...
service-mode?         false                           false
diag-script           none                            normal
diag-level            off                             max
diag-switch?          false                           false
error-reset-recovery  sync                            sync
```

OpenBoot `probe-scsi` and `probe-scsi-all` Commands

To diagnose problems with the SCSI or FC-AL devices, you can use the OpenBoot `probe-scsi` and `probe-scsi-all` commands. Both commands require that you get to the `ok` prompt after a reset.

Note – When it is not practical to halt the system, you can use SunVTS software as an alternative method of testing the SCSI and FC-AL interfaces. See "About SunVTS Software" on page 38 for more information.

The `probe-scsi` command transmits an inquiry command to all SCSI and FC-AL devices connected to the on-board SCSI and FC-AL controllers. This includes any internal tape or DVD/CD-ROM drives connected to an optional SCSI controller. For any SCSI or FC-AL device that is connected and active, its target address, unit number, device type, and manufacturer name are displayed.

Note – You can also use the `probe-scsi` command to isolate failures on the FC-AL loop. See "FC-AL Loop or Disk Drive Failure" on page 80 for more information.

The `probe-scsi-all` command transmits an inquiry command to all SCSI and FC-AL devices connected to the on-board SCSI and FC-AL controllers, and any host adapters installed in PCI slots. The first identifier listed in the display is the host adapter address in the system device tree, followed by the device identification data.

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

```
ok probe-scsi
/pci@8,600000/SUNW,q1c@2
LiD HA LUN  --- Port WWN ---  ----- Disk description -----
 0   0   0  2100002037bd356f  SEAGATE ST318304FSUN18G 042D
 6   6   0  508002000011fd5d  SUNW    SUNWGS INT FCBPL9216
 8   8   0  2100002037bd3981  SEAGATE ST318304FSUN18G 042D
```

About Diagnosing Specific Problems

This section describes how to diagnose the following problems:

Network Communication Failure

Symptom

The system is unable to communicate over the network.

Action

Your system conforms to the Ethernet 10/100BASE-T standard, which states that the Ethernet 10BASE-T link integrity test function should always be enabled on both the host system and the Ethernet hub. If you have trouble establishing a connection between the Sun Fire V890 server and your Ethernet hub, verify that the Ethernet hub also has the link test function enabled.

This problem applies only to 10BASE-T network hubs, where the Ethernet link integrity test is optional. This is not a problem for 100BASE-T networks, where the test is enabled by default. Refer to the documentation provided with your Ethernet hub for more information about the link integrity test function.

Use the `test` command to test an individual network device. At the `ok` prompt, type `test` and the full path name of the device as shown in the following example:

```
ok test /pci@9,700000/network@1,1
```

If you connect the system to a network and the network does not respond, use the OpenBoot PROM command `watch-net-all` to display conditions for all network connections:

```
ok watch-net-all
```

For most PCI Ethernet cards, the link integrity test function can be enabled or disabled with a hardware jumper on the PCI card, which you must set manually. (See the documentation supplied with the card.) For the standard TPE I/O board port, the link test is enabled or disabled through software, as described below.

Note – Some hub designs permanently enable or disable the link integrity test through a hardware jumper. In this case, refer to the hub installation or user manual for details of how the test is implemented.

Determining the Device Name of the Ethernet Interface

To enable or disable the link integrity test for the standard Ethernet interface, or for a PCI-based Ethernet interface, you must first know the device name of the desired Ethernet interface. To list the device name, follow these steps:

1. **Shut down the operating system and take the system to the `ok` prompt.**
2. **Determine the device name for the desired Ethernet interface:**
   a. **Type:**

```
ok show-devs
```

   b. **In the `show-devs` listing, find the device name for the desired Ethernet interface.**

      The device name is `/pci@9,700000/network@1,1` for the Fast Ethernet interface. For a PCI-based Ethernet interface, the device name may appear similar to the following: `/pci@8,700000/pci@2/SUNW,hme@0,1`

Enabling or Disabling the Link Integrity Test

Solution 1

Use this method while the operating system is running:

1. **Become superuser.**
2. **Type:**

```
# eeprom nvramrc="probe-all install-console banner apply
disable-link-pulse device-name"
  (Repeat for any additional device names.)
# eeprom "use-nvramrc?"=true
```

3. **Reboot the system (when convenient) to make the changes effective.**

Solution 2

Use this alternative method when the system is already at the OpenBoot prompt:

1. **At the** `ok` **prompt, type:**

```
ok nvedit
0: probe-all install-console banner
1: apply disable-link-pulse device-name
(Repeat this step for other device names as needed.)
(Press CONTROL-C to exit nvedit.)
ok nvstore
ok setenv use-nvramrc? true
```

2. **Reboot the system to make the changes effective.**

Power-On Failure

Symptom

The system attempts to power on but does not boot or initialize the terminal or monitor.

Action

1. **Verify that the CPU/Memory boards are seated correctly.**
2. **Run POST diagnostics.**

   See "Running POST Diagnostics" on page 8.
3. **Observe POST results.**

   Check the POST output using a locally attached terminal, `tip` connection, or RSC console. If you see no front panel LED activity, a power supply may be defective. See the *Sun Fire V890 Server Owner's Guide* for information about power supply LED indications.

   If the front panel System Fault LED remains lit or the POST output contains an error message, POST has failed. The most probable cause for this type of failure is the motherboard.
4. **Before you replace the motherboard, run the OpenBoot Diagnostics `test-all` command from the `ok` prompt or `obdiag>` prompt.**

Note – To get to the `ok` prompt, you must set the OpenBoot PROM configuration variable `auto-boot?` to `false` and then reset the system. (The default setting for `auto-boot?` is `true`.) See "Running OpenBoot Diagnostics" on page 31 for instructions.

```
ok test-all
```

5. **If OpenBoot Diagnostics error messages show any defective components, remove or replace those components and run firmware diagnostics again.**

   Remove any failed components that are optional. Replace any failed components that are required for a minimum configuration. Be sure the required eight DIMMs are installed in groups A0 and B0 for each CPU/Memory board installed.
6. **If POST still fails after you have removed or replaced all failed components, replace the motherboard.**

Video Output Failure

Symptom

No video at the system monitor.

Action

1. **Check that the power cord is connected to the monitor and to the wall outlet.**
2. **Verify with a volt-ohmmeter that the wall outlet is supplying AC power.**
3. **Verify that the video cable connection is secure between the monitor and the video output port.**

   Use a volt-ohmmeter to perform the continuity test on the video cable.
4. **If the cables are connected securely, troubleshoot the monitor and the graphics card. Use the `test` command.**

```
ok test screen
```

Note – To test the graphics card, a graphics display may be required.

RSC Console Failure

Symptom

The system console has been redirected to an RSC console, but the RSC console is not working.

Action

The most likely cause of this problem is a faulty system controller card. To recover from this problem and gain access to the system from a local system console, follow these steps:

1. **Press the system Power button briefly to initiate a graceful software shutdown.**

2. **Make sure that the system is connected to a local console device.**

   Install a local console if necessary. See the *Sun Fire V890 Server Owner's Guide* for instructions.

3. **Press and release the Power button and wait until the System Fault LED on the front panel begins to blink.**

4. **Immediately press the Power button twice (with a one-second delay between presses).**

   A screen similar to the following is displayed to indicate that you have successfully reset the OpenBoot NVRAM configuration variables to their default values.

```
Sun Fire V890 (8 X UltraSPARC-III), Keyboard Present
OpenBoot x.x, 256 MB memory installed, Serial #xxxxxxxx.
Ethernet address xx:xx:xx:xx:xx:xx, Host ID: xxxxxxxx.


Safe NVRAM mode, the following nvram configuration variables have
been overridden:
  'diag-switch?' is false
  'use-nvramrc?' is false
  'input-device', 'output-device' are defaulted
  'ttya-mode', 'ttyb-mode' are defaulted

These changes are temporary and the original values will be
restored after the next hardware or software reset.

ok
```

   By changing the NVRAM configuration variables to their default values, you *temporarily* redirect the system console to the local console device. Note that these NVRAM settings are reset to the defaults *for this power cycle only.* If you do nothing other than reset the system at this point, the values are not permanently changed. Only settings that you change manually at this point become permanent.

5. **To permanently redirect the system console to the local console device, type the following commands at the system** `ok` **prompt:**

```
ok diag-console ttya
ok setenv input-device keyboard
ok setenv output-device screen
```

6. **To cause the changes to take effect, power cycle the system, or type:**

```
ok reset-all
```

The system permanently stores the parameter changes

7. **Run OpenBoot Diagnostics and/or SunVTS tests for the system controller card.**
8. **Replace the system controller card, if necessary.**

FC-AL Loop or Disk Drive Failure

Symptom

A disk drive read, write, or parity error is reported by the operating system or a software application.

Action

- **Replace the drive indicated by the failure message.**

Symptom

An internal FC-AL disk drive fails to boot, is not responding to commands, or an FC-AL loop fails to initialize.

Action

Run OpenBoot Diagnostics tests for the mass storage subsystem.

1. **At the** `ok` **prompt, type:**

```
ok setenv auto-boot? false
ok setenv diag-level max
ok setenv diag-switch true
ok setenv test-args verbose,subtests
```

2. **Power off the system.**

3. **Verify that all cables attached to the FC-AL disk backplanes are properly connected.**

4. **Power on the system and observe the POST status messages.**

   If POST reports a problem, replace the component indicated by the failure message and repeat POST diagnostics until the problem is resolved.

5. **At the `ok` prompt, type:**

```
ok obdiag
```

   The OpenBoot Diagnostics menu is displayed, followed by the `obdiag>` prompt.

```
ok obdiag
 ____________________________________________________________________________
|                                o b d i a g                                 |
|____________________________ ____________________________ ____________________|
|                            |                            |                    |
|  1 SUNW,qlc@2              |  2 bbc@1,0                 |  3 bbc@1,500000    |
|  4 controller@0,16         |  5 controller@0,1a         |  6 ebus@1          |
|  7 flashprom@0,0           |  8 gpio@1,300600           |  9 hotplug-controller@0,|
| 10 hotplug-controller@0,   | 11 hotplug-controller@0,   | 12 hotplug-controller@0,|
| 13 i2c@1,2e                | 14 i2c@1,30                | 15 i2c@1,50002e    |
| 16 i2c@1,500030            | 17 ide@1                   | 18 network@1       |
| 19 network@1,1             | 20 pmc@1,300700            | 21 rsc-control@1,3062f8 |
| 22 rtc@1,300070            | 23 serial@1,400000         | 24 usb@1,3         |
|____________________________|____________________________|____________________|
|       Commands: test test-all except help what setenv set-default exit     |
|____________________________________________________________________________|
|                    diag-passes=1 diag-level=max test-args=                 |
|____________________________________________________________________________|

obdiag>
```

6. **Test segment 5 of the I^2C bus (`i2c@1,30`) to verify that it is operating correctly.**

   Enter the test number corresponding to the `i2c@1,30` test. For example:

```
obdiag> test 14
```

Note – The OpenBoot Diagnostics menu entries and test numbers vary according to system configuration.

I^2C segment 5 must be working correctly in order to test the FC-AL subsystem. If this test fails, test the remaining segments of the I^2C bus and replace the component or components indicated by the failure messages. Segment 5 test failures can also result from a faulty I^2C cable.

7. **Run the SSC-100 SES controller tests in the following order:**

   a. `controller@0,16` **– base backplane Loop A**

   b. `controller@0,1c` **– expansion backplane Loop A (if installed)**

   c. `controller@0,1a` **– base backplane Loop B**

   d. `controller@0,1e` **– expansion backplane Loop A (if installed)**

   - If the tests indicate a problem with any of the following components—DPM, CRC, SSC-100, SSC-050, or LM75—the most likely source of the problem is the backplane under test. Replace the backplane and repeat the test.
   - If a `loop-empty` subtest fails in a single backplane configuration, replace the backplane and repeat the test.
   - If a `loop-empty` subtest fails in a dual-backplane configuration, remove the FC-AL data cables between backplanes and repeat the test. If the failure persists, replace the backplane under test; otherwise, the failure may be due to the other backplane or the FC-AL cables between the two.
   - If a failure message identifies one or more specific disks, replace the disks with known good disks and repeat the test.

8. **Run the ISP2200A FC-AL controller tests in the following order:**

   a. `SUNW,qlc@2` **– on-board FC-AL controller (Loop A)**

   b. `SUNW,qlc@4` **– PCI FC-AL controller (Loop B, if installed)**

   For example:

```
ok test 1,2
```

   If a failure message identifies one or more specific disks, replace the disks with known good disks and repeat the testing. Disk failure messages identify a specific disk by its AL_PA address, according to the following table.

Base Backplane	AL_PA	Expansion Backplane	AL_PA
Disk 0	EF	Disk 6	D9
Disk 1	E8	Disk 7	D6
Disk 2	E4	Disk 8	D5
Disk 3	E2	Disk 9	D4
Disk 4	E1	Disk 10	D3
Disk 5	E0	Disk 11	D2
SSC-100 SES processor	DC		

Other types of failures during the on-board controller test usually indicate a problem with the motherboard or the motherboard FC-AL cable. When testing the PCI controller, these types of failure messages point to the PCI card or the FC-AL cable between the card and the base backplane.

In a dual-backplane configuration, removing the FC-AL cables between backplanes and repeating the test can help to isolate the problem.

DVD-ROM Drive Failure

Symptom

A DVD-ROM drive read error or parity error is reported by the operating system or a software application.

Action

- **Replace the DVD-ROM drive.**

Symptom

DVD-ROM drive fails to boot or is not responding to commands.

Action

Test the drive response to the `probe-ide` command as follows.

Note – You must halt the system to execute the `probe-ide` command. If this is not practical, you can use the SunVTS software to test the DVD-ROM. See "About SunVTS Software" on page 38.

---

1. **At the `ok` prompt, type:**

```
ok setenv auto-boot? false
ok reset-all
ok probe-ide
```

2. **Check the output message.**

   If a target address, unit number, device type, and manufacturer name are displayed for the device, the system IDE controller has successfully probed the device. This indicates that the motherboard is operating correctly.

3. **Take one of the following actions, depending on what the** `probe-ide` **command reports:**

   a. **Replace the DVD-ROM data cable.**

   b. **If the problem is still evident after replacing the cable, replace the drive.**

   c. **If the problem is still evident, replace the motherboard.**

Power Supply Failure

Symptom

If there is a problem with a power supply, the environmental monitoring system lights the following LEDs:

- System Fault LED on the front panel
- Power Fault LED on the status and control panel
- Fault LED at the rear of the problem power supply

In addition, the AC Status and DC Status LEDs at the rear of each power supply indicate any problem with the AC input and DC output, respectively. See the *Sun Fire V890 Server Owner's Guide* for more information about the LEDs.

Action

- **After you identify the problem power supply, replace it according to the removal and installation instructions in the** *Sun Fire V890 Server Service Manual***.**

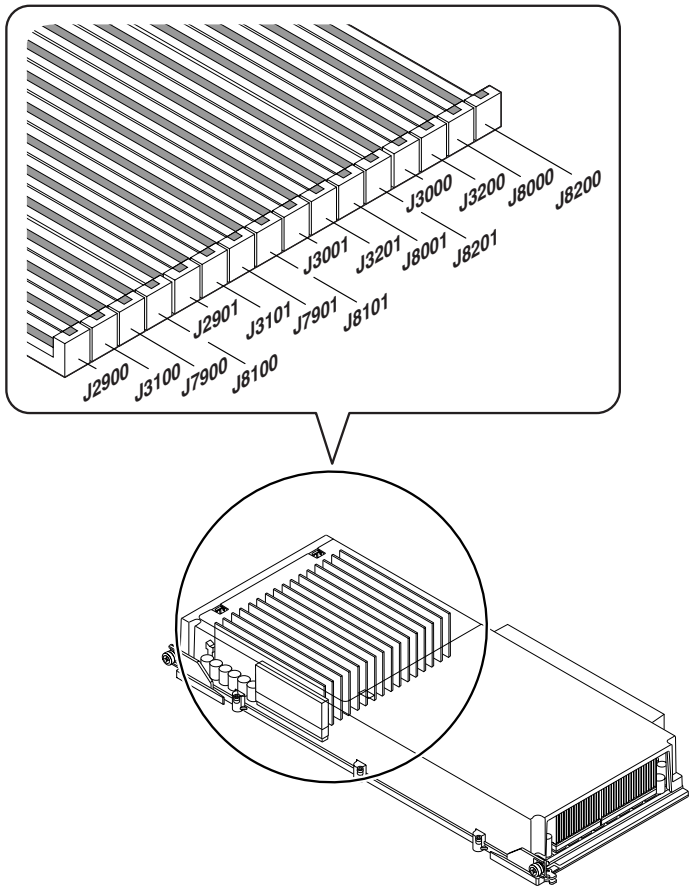
DIMM Failure

Symptom

SunVTS and POST diagnostics can report memory errors encountered during program execution. Memory error messages typically indicate the location number ("J" number) of the failing DIMM.

Action

1. **Use the following diagram to identify the location of a failing DIMM from its J number.**



2. **After you identify the defective DIMM, replace it according to the removal and installation instructions in the** *Sun Fire V890 Server Service Manual***.**

Environmental Failures

The Sun Fire V890 server features an environmental monitoring subsystem designed to protect against:

- Extreme temperatures
- Lack of adequate airflow through the system
- Power supply problems

Monitoring and control capabilities reside at the operating system level as well as in the system's flash PROM firmware. This ensures that monitoring capabilities remain operational even if the system has halted or is unable to boot.

The environmental monitoring subsystem uses an industry-standard I^2C bus. The I^2C bus is a simple two-wire serial bus, used throughout the system to allow the monitoring and control of temperature sensors, fans, power supplies, status LEDs, and the front panel keyswitch.

Monitoring Temperatures

Temperature sensors are located throughout the system to monitor the ambient temperature of the system and the temperature of each CPU. The monitoring subsystem frequently polls each sensor and uses the sampled temperatures to report and respond to any overtemperature or undertemperature conditions.

The hardware and software together ensure that the temperatures within the enclosure do not stray outside predetermined "safe operation" ranges. If the temperature observed by a sensor falls below a low-temperature warning threshold or rises above a high-temperature warning threshold, the monitoring subsystem software generates a Warning message to the system console. If the temperature exceeds a low-temperature or high-temperature critical threshold, the software issues a Critical message and proceeds to gracefully shut down the system. In both cases, the System Fault and Thermal Fault LEDs on the front status panel are illuminated to indicate the nature of the problem.

This thermal shutdown capability is also built into the hardware circuitry as a fail-safe measure. This feature provides backup thermal protection in the unlikely event that the environmental monitoring subsystem becomes disabled at both the software and firmware levels.

All error and warning messages are displayed on the system console (if one is attached) and are logged in the `/var/adm/messages` file. Front panel fault LEDs remain lit after an automatic system shutdown to aid in problem diagnosis.

Monitoring Airflow

The monitoring subsystem is also designed to detect fan failures. The basic system features three primary fan trays, which include a total of five individual fans. Systems equipped with the redundant cooling option include three additional (secondary) fan trays for a total of 10 individual fans. During normal operation, only the five primary fans are active.

If any primary fan fails, the monitoring subsystem detects the failure and performs the following:

- Generates an error message and logs it in the `/var/adm/messages` file
- Lights the System Fault and Thermal Fault LEDs on the status and control panel
- Lights the appropriate Fan Fault LED inside the system
- Automatically activates the appropriate secondary fan tray (if installed)

Monitoring Power Supplies

The power subsystem is monitored in a similar fashion. The monitoring subsystem periodically polls the power supply status registers for a power supply OK status, indicating the status of each supply's 3.3V, 5.0V, 12V, and 48V DC outputs.

If a power supply problem is detected, an error message is displayed on the system console and logged in the `/var/adm/messages` file. The System Fault and Power Fault LEDs on the status and control panel are also lit. LEDs located on the back of each power supply indicate the source and nature of the fault.

Note – The Sun Fire V890 server power supplies have their own built-in overtemperature protection circuits that will automatically shut down the supplies in response to certain overtemperature and power fault conditions. To recover from an automatic power supply shutdown, you must disconnect the AC power cord, wait approximately 10 seconds, and then reconnect the power cord.

Environmental Error Messages

The error messages, generated by the monitoring subsystem in response to an environmental error condition are listed and described in the following table. The environmental error messages are displayed on the system console (if one is attached) and logged in the `/var/adm/messages` file.

Message	Type	Description
`CRITICAL: HIGH TEMPERATURE DETECTED` *Temp*, *Temperature-Sensor*	Critical	Indicates that the temperature measured at *Temperature-Sensor* has exceeded the critical threshold. This message is displayed briefly and then followed by the shutdown message, "`The system will be shutting down in one minute`." After one minute, the system automatically shuts down. **Note:** Output from the `prtpicl -v` command shows the high-temperature critical threshold for each component. See the `prtpicl` man page for information.
`CRITICAL: LOW TEMPERATURE DETECTED` *Temp*, *Temperature-Sensor*	Critical	Indicates that the temperature measured at *Temperature-Sensor* has fallen below the critical threshold. This message is displayed briefly and then followed by the shutdown message, "`The system will be shutting down in one minute`." After one minute, the system automatically shuts down. **Note:** Output from the `prtpicl -v` command shows the low-temperature critical threshold for each component. See the `prtpicl` man page for information.
`WARNING: HIGH TEMPERATURE DETECTED` *Temp*, *Temperature-Sensor*	Warning	Indicates that the temperature measured at *Temperature-Sensor* has exceeded the warning threshold. If the temperature continues to rise and exceeds the critical threshold, the system issues the "`CRITICAL: HIGH TEMPERATURE...`" Warning and the shut down message. **Note:** Output from the `prtpicl` command shows the high-temperature warning threshold for each component. See the `prtpicl` man page for information.

Message	Type	Description
`WARNING: LOW TEMPERATURE DETECTED` *Temp , Temperature-Sensor*	Warning	Indicates that the temperature measured at *Temperature-Sensor* has fallen below the warning threshold. If the temperature continues to fall and goes below the critical threshold, the system issues the "`CRITICAL: LOW TEMPERATURE...`" warning and the shutdown message. **Note:** Output from the `prtpicl` command shows the low-temperature warning threshold for each component. See the `prtpicl` man page for information.
`WARNING: Device` *Device* `failure detected`	Warning	Indicates that there is a problem with a power supply or fan. The system may shut down abruptly if *Device* identifies a power supply or fan in a non-redundant configuration. **Note:** PS0 is the right-side power supply; PS1 is the center power supply; PS2 is the left-side power supply.
`WARNING: Fan missing, id =` *Fan*	Warning	Appears at boot time if a primary fan tray is missing. The missing fan tray is identified by the value *Fan*.

`WARNING: Power supply overcurrent detected`
 	
`WARNING: Only 1 Power Supply in system ADD 2nd Power Supply`
 	
`WARNING: Power Supply at 95% current Remove some load`
	Warning	Indicates a power supply is overloaded. The "`Power supply overcurrent detected`" message appears with one of the other warning messages ("`...Add 2nd Power Supply`" "`...Remove some load`" message).

`WARNING: Secondary fan failure, device` *Device*	Warning	Indicates a secondary fan is turned on and the speed of the fan is zero. Secondary fans are turned on only if the primary fans are not present or not operational.
`Device` *Device* `inserted`	Advisory	Appears as a hot-swap message indicating that a power supply or fan identified by *Device* was installed without service disruption.
`Device` *Device* `removed`	Advisory	Appears as a hot-swap message indicating that a power supply or fan identified by *Device* was removed without service disruption.
`Device` *Device* `OK`	Advisory	Appears when a power supply or fan failure reported by the message "`WARNING: Device failure detected`" is corrected.

Message	Type	Description

`Device` *Power-Supply* `unplugged`
 	
`Device` *Power-Supply* `plugged in`
	Advisory	Indicates a power supply is inserted, but the AC power cord is not plugged in. As soon as the AC cord is plugged in, the message "`Device POWER SUPPLY plugged in`" is displayed.
		
		Note - Environmental monitoring of a power supply occurs only if the power cord is plugged in.

Disk `Error Reported` *Disk* `Error Cleared`	Advisory	Appears if a fault is detected for any of the installed internal disks. The message "*Disk* `Error cleared`" appears when the disk fault is cleared.
`Keyswitch position changed to` *Position*	Advisory	Indicates keyswitch position has changed and gives the current position.

Index