SunVTS™ 5.1 Patch Set 2 Documentation Supplement

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054 U.S.A.
650-960-1300

Part No. 817-0937-10
April 2003, Revision A

Send comments about this document to: docfeedback@sun.com

Please
Recycle

Adobe PostScript

Contents

CHAPTER 1

Introduction

This manual is a supplement to the SunVTS™ 5.1 documentation and describes new features, tests, and test enhancements that are developed in the SunVTS 5.1 Patch Set releases. The new features, tests, and test enhancements included in this document are provided in the SunVTS 5.1 Patch Set 2 (PS2) software that is distributed on the Solaris Software Supplement CD.

For overall SunVTS features, test configuration modes, interfaces, and options refer to the *SunVTS 5.1 User's Guide*. Refer to the *SunVTS 5.1 Test Reference Manual* for detailed information on SunVTS test software and the full collection of tests released with SunVTS 5.1.

The new features of the SunVTS software included in SunVTS 5.1 PS2 are described in Chapter 2.

The following new tests are introduced in this release:

- Alarm Card 2 Test (`alarm2test`), described in Chapter 3.
- Sun™ XVR-1200 Graphics Accelerator Test (`jfbtest`), described in Chapter 4.
- Sun™ XVR-100 Graphics Accelerator Test (`pfbtest`), described in Chapter 5.
- Sun™ XVR-4000 Graphics Accelerator Test (`zulutest`), described in Chapter 6.

The following previously released tests are enhanced in this release:

- Disk and Floppy Drives Test (`disktest`), described in Chapter 7.
- System Test (`systest`), described in Chapter 8.
- Virtual Memory Test (`vmemtest`), described in Chapter 9.

The following tests were introduced in the SunVTS 5.1 PS1 release:

- Blade Support Chip Test (`bsctest`), described in Chapter 10.
- Environmental Test (`env6test`), described in Chapter 11.
- I2C Inter-Integrated Circuit Test (`i2c2test`), described in Chapter 12.

The following previously released tests were enhanced in the SunVTS 5.1 PS1 release:

- Integer Unit Test (`iutest`), described in Chapter 13.
- System Service Processor Test (`ssptest`), described in Chapter 14.

Note – The System Service Processor test (`ssptest`) was previously titled the Remote System Control test (`rsctest`) in SunVTS 5.1. The reason for this change is that this test now supports Advanced Lights-Out Management hardware in addition to both Remote System Control 1.0 and 2.0 hardware.

CHAPTER 2

New SunVTS Features

This chapter describes new features and user interface enhancements that are developed in Patch Set releases of SunVTS software.

Schedule Manager

This section describes how to use the Schedule Manager, which is a new feature introduced in SunVTS 5.1 Patch Set 2 (PS2). The Schedule Manager is configured for use with the SunVTS Common Desktop Environment (CDE) graphical user interface. For details on how to start this interface, refer to the *SunVTS 5.1 User's Guide*.

The Schedule Manager allows you to create schedules to perform tests at a specific time, periodically, at intervals, or on kernel idle. You can configure schedules to perform tests with standard options or an option file. If errors occur when running a schedule, the Schedule Manager indicates the cause.

The Schedule Manager allows you to configure a schedule to switch from one test mode to another at a specific time. Additionally, you can specify the duration to run the schedule or specify unlimited, the default is one hour.

Schedules can be performed in the standard SunVTS kernel state. If the SunVTS kernel is not running, the Schedule Manager can invoke a SunVTS kernel session at a scheduled time. With the Schedule Manager you can also force a start of a schedule and stop the currently running schedule at a specified time.

Configuring Schedules

The Schedule Manager allows you to create, edit, and delete schedules. Creating a new schedule requires selecting standard options or an option file. You must save schedules with a unique name. If you try to save a new schedule or modify an

existing one with a name already in use, the Schedule Manager will prompt you before overwriting. You can also view the details of all of the schedules in the CDE user interface.

Configuring Schedules With the SunVTS CDE User Interface

From the main SunVTS Diagnostic window, select *Scheduler→Schedule Manager→Create Schedule* to bring up the Schedule Manager dialog box for creating, editing, or deleting schedules. You can also remove all currently running schedules by selecting *Scheduler→Schedule Manager→Clean All* from the main window. The main SunVTS Diagnostic window indicates the currently running schedule.

To reach the Schedule Manager dialog box below, select *Scheduler→Schedule Manager→Create Schedule* from the main SunVTS Diagnostic window. A list of the existing schedules and the standard options are displayed.

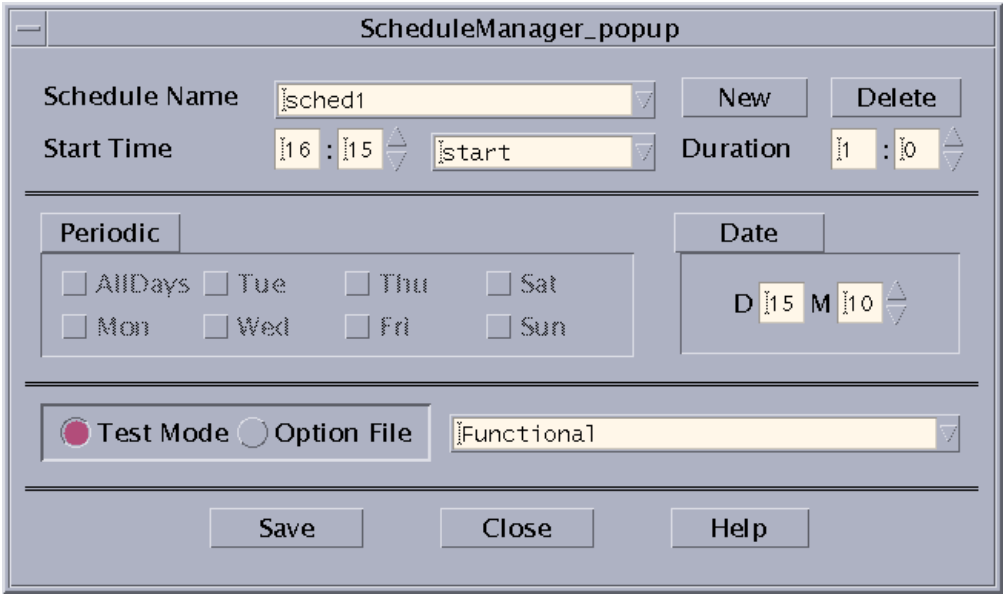

FIGURE 2-1 Schedule Manager Dialog Box

The following table describes the options listed in the Schedule Manager dialog box:

TABLE 2-1 Schedule Manager Options

CDE Inerface Options	Description
Schedule Name	Allows you to enter the name for which the current schedule will be saved. Also allows you to select and bring up existing schedules for editing from the drop-down menu.
New	Creates a new schedule with the default options.
Delete	Deletes the selected schedule that is registered with the kernel.
Start Time	Allows you to specify the time for a schedule to start.
Duration	Allows you to specify the duration to run the schedule. The default is one hour.
Periodic	Allows you to run the schedule periodically and select what days of the week you want to run the schedule. You can also select *All Days*.
Date	Allows you to specify a date that you want the schedule to start.
Test Mode or Option File	Selecting Test Mode allows you to specify which SunVTS test mode you want to use in the drop-down menu. Selecting Option File allows you to select an option file from the drop-down menu.
Save	Saves and registers the schedule to the kernel and `crontab`.
Clean All	This option can be selected from the main SunVTS Diagnostic window. This option removes all running schedules at the specified time.

To bring up an existing schedule for viewing or editing, select a schedule in the *Schedule Name* drop-down menu, or enter the name of a schedule in the field.

CHAPTER 3

Alarm Card Test for Netra™ CT Systems (`alarm2test`)

The `alarm2test` exercises the Alarm Card and System Controller Board on the Sun Netra™ CT 410 and CT 810 systems.

The Alarm Card is a hot-swappable add-on option for the Netra CT systems which provides secure remote access for system monitoring, failure recovery, and alarm notification. The Alarm Card can be used in both front and rear-access systems.

This test is not scalable.

Note – The Netra CT 410/810 system only runs the 64-bit OS (to take full advantage of UltraSPARCII). Although, `alarm2test` is available in 32-bit and 64-bit mode, only the 64-bit version of `alarm2test` can be performed on a Netra CT 410/810 system.

`alarm2test` Requirements

Solaris 9 4/03 operating environment or later is required to perform the `alarm2test`. Ethernet loopback and serial loopback connectors are also required to perform the `alarm2test`. In addition, you are required to select the Intervention mode due to the serial and Ethernet loopback connectors.

`alarm2test` Subtests

`alarm2test` consists of eight subtests which test and report on the following:

- Ethernet Internal, External, PHY loopback and PING test
- Internal/External loopback test on serial ports
- Checksum test on FLASH
- Alarm relay on/off test
- System status panel LED test
- Fan status test
- Power supply test
- FruID checksum test

`alarm2test` Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

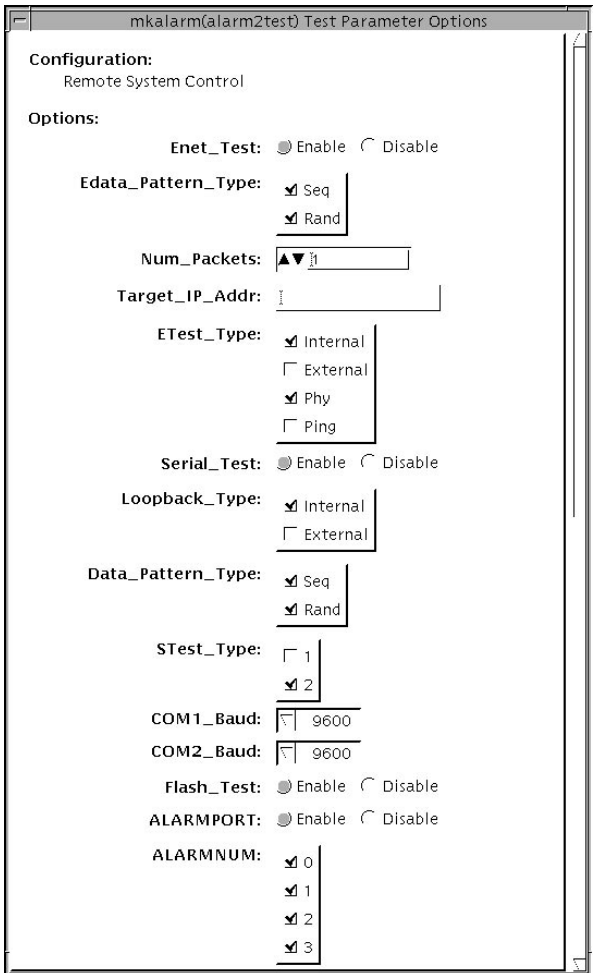

FIGURE 3-1 `alarm2test` Test Parameter Options Dialog Box With the Scroll Bar at the Top

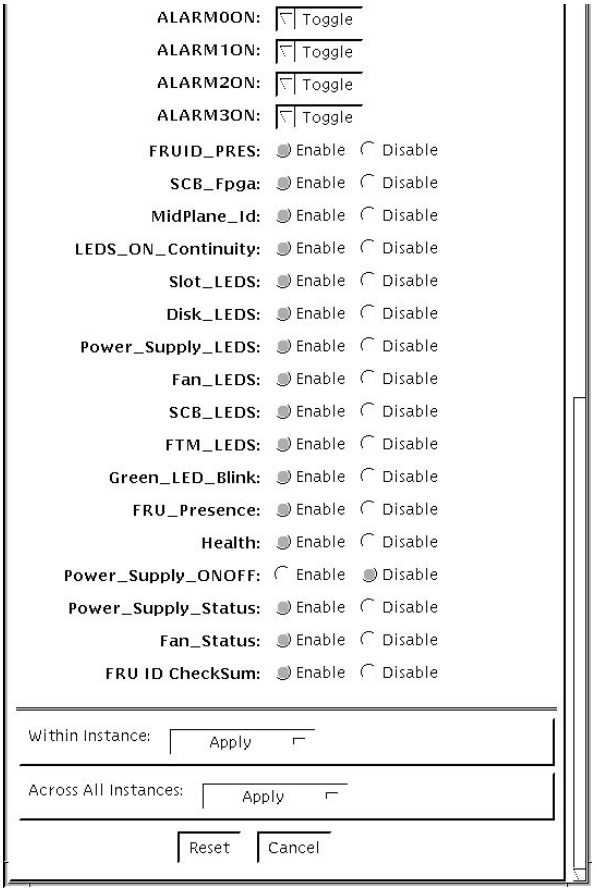

FIGURE 3-2 `alarm2test` Test Parameter Options Dialog Box With the Scroll Bar at the Bottom

TABLE 3-1 `alarm2test` Options

Option	Description
Enet_Test	Enables or disables Ethernet testing.
Edata_Pattern_Type	Selects the type of data pattern for `Enet_Test`: Sequential, Random, or both.
Num_Packets	Specifies the number of data packets to be sent in one test loop.
Target_IP_Addr	Specifies the IP address of a host to use for the ping test.

TABLE 3-1 `alarm2test` Options

Option	Description
Etest_Type	Selects any or all internal, external, Phy (ethernet transceiver), or ping tests.
Serial_Test	Enables or disables `serial_test`.
Loopback_Type	Selects internal loopback, external loopback, or both.
Data_Pattern_Type	Selects the type of data pattern for `serial_test`: Sequential, Random, or both.
STest_Type	Selects ports to be tested: `c`, `d`, `u`, or `v`.
COM1_Baud	Selects the alarm card's COM1 port baud rate.
COM2_Baud	Selects the alarm card's COM2 port baud rate.
Flash_Test	Enables or disables the flash checksum test.
ALARMPORT	Enables or disables the alarmport test.
ALARMNUM	Selects any or all alarm ports to be tested: 0, 1, 2, 3.
ALARM0ON	Turns on, turns off, or toggles (on then off) alarm port 0.
ALARM1ON	Turns on, turns off, or toggles (on then off) alarm port 1.
ALARM2ON	Turns on, turns off, or toggles (on then off) alarm port 2.
ALARM3ON	Turns on, turns off, or toggles (on then off) alarm port 3.
FRUID_PRES	Enable/Disable FRU ID Presence test
SCB_Fpga	Enable/Disable scb_fpga register test
MidPlane_Id	Enable/Disable Midplane ID test
LEDS_ON_Continuity	Enable/Disable SCB LEDs test
Slot_LEDS	Enable/Disable Slot LEDs test
Disk_LEDS	Enable/Disable Disk LEDs test
Power_Supply_LEDS	Enable/Disable Power Supply LEDs test
Fan_LEDS	Enable/Disable Fan LED test
SCB_LEDS	Enable/Disable SCB Resgister LEDs test
FTM_LEDS	Enable/Disable Front Tranission Module LEDs test
Green_LED_Blink	Enable/Disable Green LED Blink test
FRU_Presence	Enable/Disable FRU Presence test
Health	Enable/Disable Health test
Power_Supply_ONOFF	Enable/Disable Power Supply On/Off test

TABLE 3-1 `alarm2test` Options

Option	Description
Power_Supply_Status	Enable/Disable Power Suppluy Status test
Fan_Status	Enable/Disable Fan Status test
FRU ID CheckSum	Enable/Disable FRU ID Checksum test for Midplane, SCB, Alarm, Fan1/Fan2, and Power Supply1/Supply2

`alarm2test` Loopbacks

The loopback tests use the following external loopbacks:

- Ethernet loopback test—standard RJ-45 connector. Connect pin 1 to pin 3, and pin 2 to pin 6.
- Serial loopback test—RJ-45. Connect pin 6 to pin 3, pin 1 to pin 8, and pin 2 to pin 7.

`alarm2test` Test Modes

TABLE 3-2 `alarm2test` Supported Test Modes

Test Mode	Description
Connection	Reports the status of the alarm card.
Functional	Runs the the full set of subtests.

`alarm2test` Command-Line Syntax

/opt/SUNWvts/bin/alarm2test *standard_arguments* **-o enet=** *E(nable)/D(isable)*, **epattype=***seq+rand*, **target=***IP_Address*, **etest=** *Internal+External+Ping+Phy*, **serial=***E(nable)* | *D(isable)*, **slb=***Internal+External*, **spattype=***Seq+Rand*, **com1baud=** *ALL* | *1200* | *2400* | *4800* | *9600* | *19200* | *38400* | *56000*, **com2baud=** *ALL* | *1200* | *2400* | *4800* | *9600* | *19200* | *38400* | *56000*, **flash=***E(nable)* | *D(isable)*, **aport=***E(nable)* | *D(isable)*, **anum=***0+1+2+3*, **a0on=***On* | *Off* | *Toggle*, **a1on=** *On* | *Off* | *Toggle*, **a2on=***On* | *Off* | *Toggle*, **a3on=***On* | *Off* | *Toggle*, **FruIdPres=** *E(nable)* | *D(isable)*, **FpgaId=***E(nable)* | *D(isable)*, **MidPlaneId=***E(nable)* | *D(isable)*,

Continuity=*E(nable)* | *D(isable)*, **SlotLeds=***E(nable)* | *D(isable)*, **DiskLeds=** *E(nable)* | *D(isable)*, **PsupplyLeds=***E(nable)* | *D(isable)*, **FanLeds=** *E(nable)* | *D(isable)*, **ScbLeds=***E(nable)* | *D(isable)*, **FtmLeds=***E(nable)* | *D(isable)*, **GreenLedsBlink=***E(nable)* | *D(isable)*, **FruPresence=***E(nable)* | *D(isable)*, **Health=***E(nable)* | *D(isable)*, **PowerSupply=***D(isable)* | *E(nable)*, **PsupplyStatus=** *E(nable)* | *D(isable)*, **FanStatus=***E(nable)* | *D(isable)*, **FruIdChkSum=** *E(nable)* | *D(isable)*

TABLE 3-3 `alarm2test` Command-Line Syntax

Argument	Explanation
enet=*E(nable)/D(isable)*	Enables or disables Ethernet testing.
epattype=*seq+rand*	Selects the type of data pattern for `Enet_Test`: Sequential, Random, or both.
target=*IP_Address*	Specifies the IP address of a host to use for the ping test.
etest= *Internal+External+Ping+Phy*	Selects any or all internal, external, Phy (ethernet transceiver), or ping tests.
serial=*E(nable)/D(isable)*	Enables or disables `serial_test`.
slb=*I+E*	Selects internal loopback, external loopback, or both.
spattype=*seq+rand*	Selects the type of data pattern for `serial_test`: Sequential, Random, or both.
com1baud=*ALL* | *specific_baud*	Defines baud rates to be used in testing the alarmcard's COM1 port.
com2baud=*ALL* | *specific_baud*	Defines baud rates to be used in testing the alarmcard's COM2 port.
flash=*E(nable)/D(isable)*	Enables or disables the flash checksum test.
aport=*[E]nable* | *[D]isable*	Enables or disables the alarmport test.
anum=*0+1+2+3*	Selects any or all alarm port to be tested: 0, 1, 2, 3
a0on=*On* | *Off/* | *T]oggle*	Turns on, turns off, or toggles (on then off) alarm port 0.
a1on=*On* | *Off* | *[T]oggle*	Turns on, turns off, or toggles (on then off) alarm port 1.
a2on=*On* | *Off* | *[T]oggle*	Turns on, turns off, or toggles (on then off) alarm port 2.
a3on=*On* | *Off* | *[T]oggle*	Turns on, turns off, or toggles (on then off) alarm port 3.
FruIdPres=*E(nable)* | *D(isable)*	Enable/Disable FRU ID Presence test
FpgaId=*E(nable)* | *D(isable)*	Enable/Disable scb_fpga register test
MidPlaneId=*E(nable)* | *D(isable)*	Enable/Disable Midplane ID test
Continuity=*E(nable)* | *D(isable)*	Enable/Disable SCB LEDs test
SlotLeds=*E(nable)* | *D(isable)*	Enable/Disable Slot LEDs test

TABLE 3-3 `alarm2test` Command-Line Syntax

Argument	Explanation *(Continued)*
`DiskLeds=`*E(nable)* | *D(isable)*	Enable/Disable Disk LEDs test
`PsupplyLeds=`*E(nable)* | *D(isable)*	Enable/Disable Power Supply LEDs test
`FanLeds=`*E(nable)* | *D(isable)*	Enable/Disable Fan LED test
`ScbLeds=`*E(nable)* | *D(isable)*	Enable/Disable SCB Resgister LEDs test
`FtmLeds=`*E(nable)* | *D(isable)*	Enable/Disable Front Tranission Module LEDs test
`GreenLedsBlink=` *E(nable)* | *D(isable)*	Enable/Disable Green LED Blink test
`FruPresence=`*E(nable)* | *D(isable)*	Enable/Disable FRU Presence test
`Health=`*E(nable)* | *D(isable)*	Enable/Disable Health test
`PowerSupply=`*D(isable)* | *E(nable)*	Enable/Disable Power Supply On/Off test
`PsupplyStatus=` *E(nable)* | *D(isable)*	Enable/Disable Power Suppluy Status test
`FanStatus=`*E(nable)* | *D(isable)*	Enable/Disable Fan Status test
`FruIdChkSum=`*E(nable)* | *D(isable)*	Enable/Disable FRU ID Checksum test for Midplane, SCB, Alarm, Fan1/Fan2, and Power Supply1/Supply2

Note – 64-bit tests are located in the `sparcv9` subdirectory `/opt/SUNWvts/bin/sparcv9/`*testname*, or the relative path to which you installed SunVTS. If a test is not present in this directory, then it may only be available as a 32-bit test. For more information refer to the "32-Bit and 64-Bit Tests" section of the *SunVTS 5.1 Test Reference Manual* (816-5145-10).

CHAPTER 4

Sun™ XVR-1200 Graphics Accelerator Test (`jfbtest`)

`jfbtest` verifies the proper functioning of the Sun™ XVR-1200 graphics accelerator.

`jfbtest` can detect and adapt to many video modes of the Sun XVR-1200 graphics accelerator. All tests can run at a resolution of 1024x768 or higher.

You can interrupt `jfbtest` using Control-C.

Test accuracy is checked using direct image comparison against compressed images. Failed pixel locations are printed as error messages.

Caution – Do not run any other application or screen saver program that uses the Sun XVR-1200 graphics accelerator port while running `jfbtest`. This combination causes SunVTS to return incorrect errors.

`jfbtest` Test Requirements

Disable all screen savers before testing any graphics device. To disable the Solaris screen saver, type the following at a UNIX prompt:

```
# xset s off
```

To turn Power Management off, type the following at a UNIX prompt:

```
# xset -dpms
```

The display resolution must be 1024x768 or higher (the standard resolution). To change resolution, go to a UNIX prompt and type:

```
# fbconfig -res 1280x1024x76
```

For full instructions on testing frame buffers, see the Testing Frame Buffers section of the *SunVTS 5.1 Test Reference Manual*.

Preparation for `jfbtest`

You should complete a few steps in advance to ensure that `jfbtest` runs as smoothly as possible.

If you are running `jfbtest` in a window system (such as CDE):

- Turn Power Management off, if it is enabled. The following is an alternate way to turn Power Management off. Change `allowFBPM=1` to `allowFBPM=0` in `/platform/sun4u/kernal/drv/jfb.conf` file.
- Make sure that no other program is running that might modify the screen during the test.
- Make sure you have permission to lock the X server. `jfbtest` is designed to lock the X server during testing to prevent screen changes.
- The CDE login window should not be displayed during testing.
- Check that the window system is only running on one Sun XVR-1200 graphics accelerator.

If you are not running `jfbtest` in a window system:

- Turn Power Management off, if it is enabled. The following is an alternate way to turn Power Management off. Change `allowFBPM=1` to `allowFBPM=0` in `/platform/sun4u/kernal/drv/jfb.conf` file.
- Make sure that no other program is running that might modify the screen during the test.
- Make sure the Sun XVR-1200 graphics accelerator being tested is not the console device. Console messages may modify the screen.

jfbtest Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

By default, all jfbtest options are enabled.

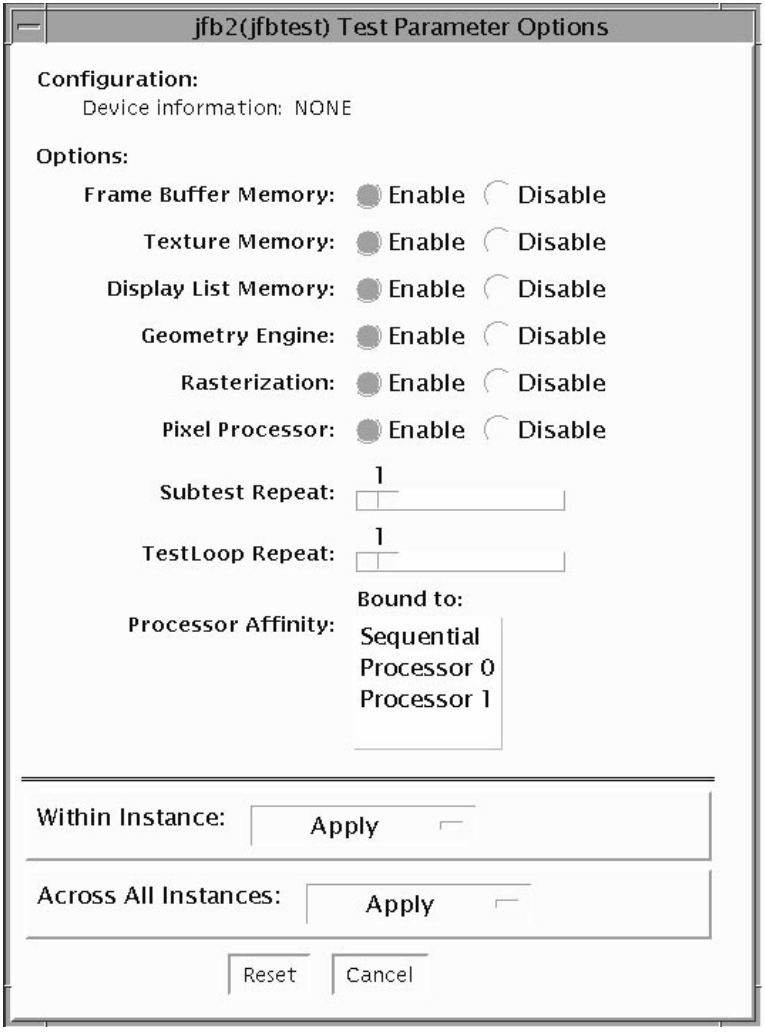

FIGURE 4-1 jfbtest Test Parameter Options Dialog Box

TABLE 4-1 `jfbtest` Options

`jfbtest` **Options**	**Description**
Frame Buffer Memory test	Thoroughly tests the Sun XVR-1200 video memory by using read and write requests. Tests for shorts or failed connections on the data bus by writing the following values to every address: • `0xFFFFFFFF` • `0xFFFF0000` • `0x0000FFFF` • `0xFF00FF00` • `0x00FF00FF` • `0xF0F0F0F0` • `0x0F0F0F0F` • `0xCCCCCCCC` • `0x33333333` • `0xAAAAAAAA` • `0x55555555` Tests for shorts or failed connections on the address bus by writing the offset of each memory location to each location and reading them back. This may also catch speed-related problems due to the volume of read/writes. Errors in the test are reported as an error in a particular address, not attributed to a specific chip. To help distinguish bit-related errors, the errors are summarized to list which bits had at least one error in the test. This test shows on the screen as random pixels.
Texture Memory test	This test is identical in process to the Frame Buffer Memory test (above). Since this test produces no visible effect, rectangles are drawn in rows across the screen to show progress.
Display List Memory test	This test is identical in process to the Frame Buffer Memory and Texture Memory tests (above), and is applied to direct burst memory. This test takes little time and no progress is displayed.
Geometry Engine test	Loads diagnostic microcode into the geometry engine and confirms that the processor operates correctly. This is a pass/fail test. This test takes little time and no progress is displayed.

TABLE 4-1 `jfbtest` Options

`jfbtest` **Options**	**Description**
Rasterization test	Renders many primitives with minimal fragment processing, to test the rasterization of the primitives. The primitives used are: • Dots • Anti-aliased dots • Lines using all for line-drawing primitives • Anti-aliased lines using all for line-drawing primitives • Triangles, Quads, and Polygons in point, line, and fill modes • Rectangles This tests for the following rasterization attributes: • Pixel coverage • Constant value registers for color, Z, and stencil • Interpolation of color, Z, and texture coordinates along lines and spans in polygons • Texture map sampling Resulting images are compared against stored images. Errors indicate which operation type and value was being tested, and the coordinate of the failed pixel.

TABLE 4-1 `jfbtest` Options

`jfbtest` **Options**	**Description**
Pixel Processor test	Tries the various pixel processing operators using a variety of fragment values. This tests the following fragment processing operations: • Depth Buffering • Blending • Alpha Test • Color Test • Color Clamp • Logic Operations • Color Matrix and Bias • Color Table • Control Planes • Fast Clear • Stencil • Scissor Clipping • Desktop Clipping • Mask Clipping • Write Masks • Window Origin • Fog • Pixel Texture • Accumulation Buffer • Pixel Buffers Resulting images are compared against stored images. Errors indicate which operation type and value was being tested and the coordinate of the failed pixel.

`jfbtest` Test Modes

Due to the nature of graphic tests, reading data from, or writing data to the frame buffer during graphic tests will disturb user operation. For this reason, `jfbtest` is only available in Functional test mode.

TABLE 4-2 `jfbtest` Supported Test Modes

Test Mode	**Description**
Functional	Runs the full set of tests.

jfbtest Command-Line Syntax

/opt/SUNWvts/bin/jfbtest *standard_arguments* **-o dev=***device_name***, fbmem=***E(nable)/D(isable)***,texmem=***E/D***,dlmem=***E/D***,geomeng=***E/D**, **rasterization=***E/D***,pixelproc=***E/D***,subtest_repeat=***number**, **test_repeat=***number*

TABLE 4-3 jfbtest Command-Line Syntax

Argument	Description
dev=*device_name*	*device_name* is the relative path name of the device being tested with respect to /dev/fbs. There is no default.
fbmem=*E/D*	Enables or disables the Frame Buffer Memory test.
texmem=*E/D*	Enables or disables the Texture Memory test.
dlmem=*E/D*	Enables or disables the Display List Memory test.
geomeng=*E/D*	Enables or disables the Geometry Engine test.
rasterization=*E/D*	Enables or disables the Rasterization test.
pixelproc=*E/D*	Enables or disables the Pixel Processing test.
subtest_repeat=*number*	Defines the number of times to repeat each subtest. The default is 1.
test_repeat=*number*	Defines the number of times to repeat a test loop before passing. The default is 1.

Note – 64-bit tests are located in the sparcv9 subdirectory: /opt/SUNWvts/bin/sparcv9/*testname, or the relative path to which you installed SunVTS*. If a test is not present in this directory, then it may only be available as a 32-bit test. For more information refer to the "32-Bit and 64-Bit Tests" section of the *SunVTS 5.1 Test Reference Manual* (816-5145-10).

CHAPTER 5

Sun™ XVR-100 Graphics Accelerator Test (`pfbtest`)

`pfbtest` tests the PCI-based Sun™ XVR-100 graphics accelerator by performing the following subtests:

- Video Memory test
- RAMDAC test
- Accelerator Port test

Caution – *Do not* run any other application or screen saver program that uses the Sun XVR-100 graphics accelerator port while running `pfbtest`. This program causes SunVTS to return incorrect errors.

Note – Disable all screen savers before testing any graphics device. Type **`xset s off`** at a UNIX prompt to disable the Solaris screen saver. Type **`xset -dpms`** (to turn off power management) or type **`xset s noblank`** (to turn off screen saver). Disable Power Management software if it is running.

Note – To start SunVTS with `vtsui`, but without `vtsk`, you must add the host name to `xhost` as: **`xhost +`** *hostname.*

For full instructions on testing frame buffers, refer to the Testing Frame Buffers section of the *SunVTS 5.1 Test Reference Manual*.

`pfbtest` Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

By default, all options are enabled except frame buffer locking.

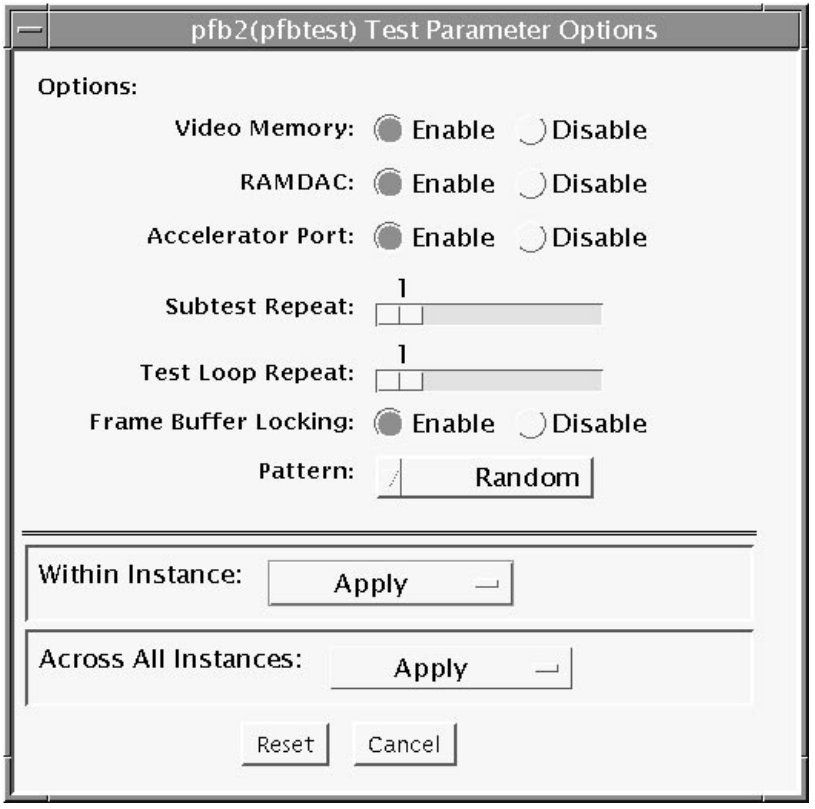

FIGURE 5-1 `pfbtest` Test Parameter Options Dialog Box

TABLE 5-1 `pfbtest` Options

`pfbtest` **Options**	**Description**
Video Memory test	Thoroughly tests the on-screen video memory (the memory part that is mapped on to the monitor) of the Sun XVR-100 graphics accelerator in 8-bit, 16-bit, 32-bit, 64-bit, and 64 byte (block) modes. Entire on-screen video memory is tested by testing 512 bit blocks at a time (8x8 pixel block). Each block is tested in two passes. Each pass consists of a data write and read. In the first pass, user specified data or random data is used, and in the second pass, one's complement of the data used in the first pass is used so that each on-screen video memory location (bit) is tested with a zero (electrical low state) and one (electrical high state).
RAMDAC test	Tests the RAMDAC in three phases. In the first phase the RAMDAC CLUT (Color LookUp Table) is tested using simple write/read patterns to determine if there are any bad bits in CLUT. The data patterns used are: • Random data • Complement of the random data (used as first data pattern) • The data pattern 0101 • The data pattern 10101 In the second phase, four different patterns are drawn on the screen. Each pattern stays on the screen for approximately 1/4 second. The four patterns are listed below. For each pattern, the signature is captured and compared with the signature obtained for the same pattern on a known good board. This test verifies that all the different data paths within the RAMDAC are functioning properly. Patterns drawn on screen: • Red ramp with cursor at top-left corner of the screen • Blue ramp with cursor at top-right corner of the screen • Green ramp with cursor at bottom-left of the screen • Grey ramp with cursor at bottom-right of the screen In the last (third) phase of the RAMDAC test the Vertical Retrace Interrupt is tested for approximately five seconds.

TABLE 5-1 `pfbtest` Options *(Continued)*

`pfbtest` **Options**	**Description**
Accelerator Port test	Tests all of the following: • Data paths (sources: fixed color, host data, blit, fixed pattern) • Arithmetic and logic unit (ALU) • Primitives (destinations: line, rectangle) • Mono to color expansion logic Primitives are drawn using a combination of different data paths (allowed), ALU functions, and color comparator functions. A checksum is generated for each data combination and is compared with the checksum generated for the same data combination on a known good board.
Frame Buffer Locking	This option is set to *disable* if the Sun XVR-100 graphics accelerator is not the console device. When the SunVTS GUI is brought up, Frame Buffer Locking is enabled by default if the Sun XVR-100 graphics accelerator is the console device. If the Sun XVR-100 graphics accelerator is not the console device, Frame Buffer Locking is disabled by default.

`pfbtest` Test Modes

Due to the nature of graphics tests, reading from or writing to the frame buffer during graphics tests will disturb user operation. This test is only available in the Functional test mode.

TABLE 5-2 `pfbtest` Supported Test Modes

Test Mode	**Description**
Functional	The `pfbtest` verifies the proper functioning of Sun XVR-100 graphics accelerator.

pfbtest Command-Line Syntax

/opt/SUNWvts/bin/pfbtest *standard_arguments* **-o dev=***device_name***, S=***subtest_number*,**F=**#*_of_subtest_loops*,**B=**#*_of_test_loops*,L=***disable*,**P=***test_pattern*

TABLE 5-3 pfbtest Command-Line Syntax

Argument	Description
dev=*device_name*	*device_name* is the relative path name of the device being tested with respect to /dev/fbs. The default is pfb0.
S=*subtest_number*	*subtest_number* is the test number of the subtest to be run. Select from the subtests below. You can run multiple subtests by adding the subtest numbers. For example, n=0x3 runs both test 1 and test 2; n=0x5 runs both test 1 and test 4. • n 0x1 VRAM • n 0x2 RAMDAC • n 0x4 Accelerator port test (Rendering Pipeline) More than one test can be selected by ORing subtest numbers. For example: n = 0x5 indicates VRAM and Rendering Pipeline tests. A hex number must be preceded by 0x, decimal numbers are also acceptable.
F=#*_of_subtest_loops*	Specifies the number of times to repeat each subtest. The default is 1.
B=#*_of_test_loops*	Specifies the number of times to repeat a test loop before passing; the default is 1.
L=*disable*	Disables the frame buffer lock. Disable the lock when the Sun XVR-100 graphics accelerator is not the console or when the server is not running on the Sun XVR-100 graphics accelerator under test.
P=*test_pattern*	Specifies the test pattern number. The default is r, for random patterns. You may also choose 0 for 0x0000000, 3 for 0x3333333, 5 for 0x5555555, or 9 for 0x9999999.

Note – 64-bit tests are located in the sparcv9 subdirectory /opt/SUNWvts/bin/sparcv9/*testname*, or the relative path to which you installed SunVTS. If a test is not present in this directory, then it may only be available as a 32-bit test. For more information refer to the "32-Bit and 64-Bit Tests" section of the *SunVTS 5.1 Test Reference Manual* (816-5145-10).

Note – Errors returned by `pfbtest` are nonspecific. It is not possible to determine which component caused a failure. In all error conditions, the field replaceable unit (FRU) is the entire Sun XVR-100 graphics accelerator.

CHAPTER 6

Sun™ XVR-4000 Graphics Accelerator Test (`zulutest`)

The `zulutest` does functional testing of the Sun™ XVR-4000 graphics accelerator device. zulutest detects and adapts to the video modes of Sun XVR-4000. All zulutest tests can be performed in several screen resolutions such as standard, Stereo, and high resolution. In Stereo mode, all tests write into the right and left eyes unless you specify otherwise. Use the fbconfig -dev device-name -prconf command to display the configuration of the frame buffer you want to test.

You can interrupt zulutest using Control-C. Turn off all other keyboard input if the CDE user interface is running on the unit being tested. Test accuracy is checked using a checksum algorithm. Possible locations of failing pixels are identified, in addition to the failing FRU.

zulutest is only available in 64-bit mode.

Caution – Do not run any 3D graphics applications screen lock or screen saver programs that uses the Sun XVR-4000 graphics accelerator port while running `zulutest`. This combination causes SunVTS to return incorrect errors.

`zulutest` Test Requirements

Disable all screen locks and screen savers before testing any graphics device. Type **`xset s off`** at a UNIX® prompt to disable the Solaris screen saver. Disable the Power Management software if it is running.

For full instructions on testing frame buffers, please refer to the Testing Frame Buffers of the *SunVTS 5.1 Test Reference Manual*.

To start SunVTS with the `vtsui`, and without the `vtsk`, you must add the host name to `xhost` as follows: `xhost + host_name`

Using `zulutest` Without X-Windows

If you perform `zulutest` on a system that was powered on without running X-Windows, you must bring up X-Windows on the Sun XVR-4000 Graphics Accelerator device under test and kill the X-Windows process before performing `zulutest`. Otherwise, the Convolve subtest will fail, and other subtests may also fail.

Note – You must enable multisampling with the `fbconfig` command before performing the following workaround. To perform `zulutest` with X-Windows (CDE) the following workaround is not necessary.

Workaround

To bring up X-Windows on the Sun XVR-4000 Graphics Accelerator device under test, enter the following command:

```
/usr/openwin/bin/Xsun -dev /dev/fbs/device_name &
```

It takes 30 to 45 seconds before Xsun comes up. To kill the Xsun process, enter the following command:

```
pkill -KILL Xsun
```

Once the Xsun process is killed, the `zulutest` can be performed without the incorrect subtest errors.

The Sun XVR-4000 Graphics Accelerator cannot perform video read back in Interlaced and Stereo modes because the Convolve subtest cannot keep up.

For `zulutest` to be able to perform the Convolve subtest, multisampling must be enabled.

`zulutest` Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

By default, all `zulutest` options are enabled except for the Stereo test.

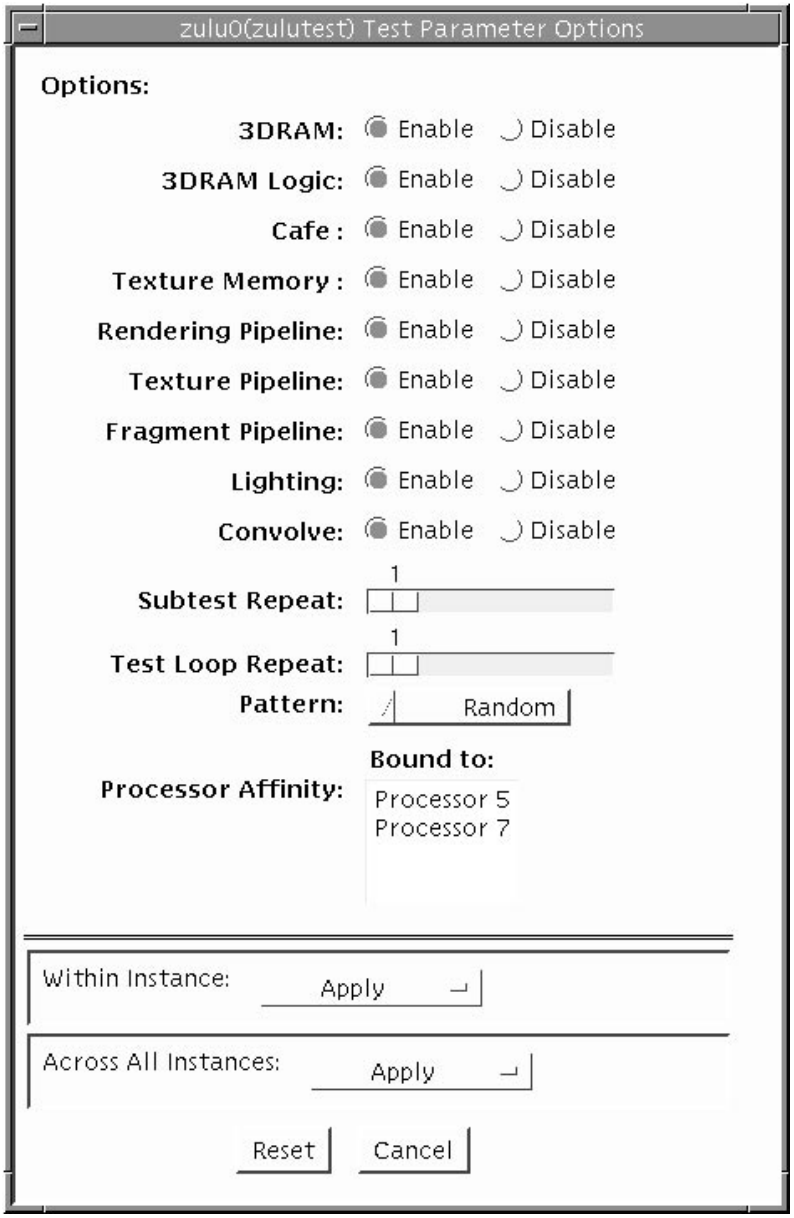

FIGURE 6-1 `zulutest` Test Parameter Options Dialog Box

TABLE 6-1 `zulutest` Options

`zulutest` **Options**	**Description**
3DRAM test	The 3DRAM test thoroughly tests the video memory in the Sun XVR-4000 graphics accelerator using 512-bit reads and writes. 3DRAM makes a full-screen pass, consisting of a write and a read to each pixel location, for each access mode on the list below. You can use either random data or specify data at the command line. A second pass is made with the one's complement of the data used in the first pass so that each memory location is tested with both a zero and a one. Errors in this subtest are attributed to the 3DRAM. A failing chip is indicated by (X, Y) locations and device-specific "U" numbers in the following access modes: • SFB Stencil 8 • SFB WID 16 • FB RGBAZ 64 - Buffer A • SFB RGBAZ 64 - Buffer B

TABLE 6-1 `zulutest` Options *(Continued)*

`zulutest` **Options**	**Description**
3DRAM Logic test	The 3DRAM Logic test provides logical functionality to the Sun XVR-4000 graphics accelerator. The following services are tested: • Compare Controls—Match AB • Compare Controls—Magnitude AB • Compare Controls—Match C • Compare Controls—Magnitude C • Match Mask—AB • Magnitude Mask—AB • Match Mask—C • Magnitude Mask—C • Raster Operations—RGB • Raster Operations—X • Raster Operations—YZ • Plane Mask—RGB Each function is tested separately with a series of SFB64 writes. A total of 16 writes are made for each different test case with Y coordinate values varying from 0 to 30 in increments of 2 pixels. This dotted column organization provides page thrashing and block flashing in all screen resolutions. For each operation, all possible combinations are tested. For example, in `ROP RGB` *new==old* there are three possible values: *new < old, new == old, and new > old*. Each of these cases are tested. Errors in this subtest are attributed to the 3DRAM.
Cafe test	This test will do non-destructive testing of the Cafe memory (RDRAM) and Cafe. Errors in this test are attributed to the Cafe and its memory.
Texture Memory test	Texture Memory test tests out all the of the Texture Memory by writing the data pattern selected (random, 0s, 1s, 5s, or 0xAs). By default, Random data is selected. The data is written using block writes and read back using block reads. Errors in this test are attributed to the Texture Memory and the Texture Memory subsystem.

TABLE 6-1 `zulutest` Options *(Continued)*

`zulutest` **Options**	**Description**
Rendering Pipeline test	Each primitive is tested thoroughly by exercising the following: • Simple Triangles • 2d primitives • 3d Primitives (such as Triangles, 3d lines etc.) • Vertex Processor Errors in this test are attributed to the pipelines of the Sun XVR-4000 graphics accelerator and/or 3DRAM.
Texture Pipeline test	This test renders textured primitives to test: • 2d texture Minification filtering • 2d texture Magnification filtering • 3d texture Minification filtering • 3d texture Magnification filtering • Texture environment • Filter4 and sharpen filters • Anisotropic filter Errors in this test are attributed to the pipelines of the Sun XVR-4000 graphics accelerator and/or 3DRAM.
Fragment Processor test	Fragment Processor test, a subtest, exercises the fragment pipe of each pipeline of the XVR-4000's. Auxiliary clipping (additive and subtractive): • Depth cueing • Alpha blend • Viewport clip (2D and 3D) • Area pattern (transparent and opaque) Errors in this test are attributed to the FBC3 and/or 3DRAM.
Lighting test	The Lighting test exercises Cafe and lighting microcode. This test lights an object with maximum number of lights that XVR-4000 can handle in hardware. A checksum is generated for the rendered image and compared with the checksum generated for the same image on a known good system. Errors in this test are attributed to the Cafe, Microcode, FBC3 and RD RAMs.

TABLE 6-1 `zulutest` Options *(Continued)*

`zulutest` **Options**	**Description**
Convolve test	Convolve test tests the Convolve chips functionality (convolution filters, Color look up tables and Gamma look up tables) along with the video read back functionality of convolves and master chip. This sub test renders an image which is made up of lines drawn radial. Then a block in the center of the image is super sampled and video read back is initiated. Once the video read back data is available to the zulutest, zulutest will generate checksum and compares with the checksum generated on known good system. Errors in this subtest can be attributed to FBC3, 3DRAM, Convolve, Master.
Stereo test	Currently, this sub test is not active. Stereo test displays an object in Stereo mode with different images for the right and left eye. You can verify proper operation by looking at the screen with stereo glasses and following the instructions displayed in the Parameter Options dialog box. This test temporarily switches the monitor into Stereo mode, renders a Stereo image and after displaying the image for five seconds, restores the monitor to its previous resolution.

`zulutest` Test Modes

Due to the nature of graphic tests, reading data from, or writing data to the frame buffer during graphic tests will disturb user operation. For this reason, `zulutest` is only available in Functional test mode.

TABLE 6-2 `zulutest` Supported Test Modes

Test Mode	**Description**
Functional	Runs the full set of tests.

zulutest Command-Line Syntax

/opt/SUNWvts/bin/sparcv9/zulutest *standard_arguments* **-o dev=** *device_name*,**S=***subtest_number*,**F=**#*_of_subtest_loops*,**B=**#*_of_test_loops*,**P=***test_pattern*

TABLE 6-3 zulutest Command-Line Syntax

Argument	Description
dev=*device_name*	*device_name* is the relative path name of the device being tested with respect to /dev/fbs; the default is zulu0.
S=*subtest_number*	*subtest_number* is the test number of the subtest to be run. Select from the subtests below. You can run multiple subtests by adding the subtest numbers together. For example, *n=0x3* runs both test 1 and test 2; *n=0x180* runs both test *0x080* and test *0x0100*. You do not need the leading zeros. • *n*—0x00001 Video Memory 3DRAM • *n*—0x00002 3DRAM Logic • *n*—0x00004 Cafe • *n*—0x00008 Texture Memory SDRAM • *n*—0x00010 Rendering Pipeline • *n*—0x00020 Texturing Pipeline • *n*—0x00040 Fragment Pipeline • *n*—0x00080 Lighting • *n*—0x00100 Convolve • *n*—0x00200 Stereo More than one subtest can be selected by ORing their subtest numbers. Example: n = 0x00011 indicates 3DRAM and Rendering Pipeline tests. A hex number must start with 0x, decimal numbers are also acceptable. [n = 0xff] If looping on a test, the verbose mode is disabled. F=n : Number of times to repeat each subtest [n = 1]. B=n : Number of times to repeat test loop before passing [n = 1]. P=*pattern* : test pattern - r for random, 0 for 0x00000000, 3 for 0x33333333, 5 for 0x55555555, or 9 for 0x9999999. [*pattern=r*]
F=#*_of_subtest_loops*	The number of times to repeat each subtest. The default is 1.
B=#*_of_test_loops*	The number of times to repeat a test loop before passing. The default is 1.
P=*test_pattern*	The test pattern number. The default is **r**, for random patterns. You may also choose 0 for 0x0000000, 3 for 0x3333333, 5 for 0x5555555, or 9 for 0x9999999.

Note – 64-bit tests are located in the `sparcv9` subdirectory `/opt/SUNWvts/bin/sparcv9/`*testname, or the relative path to which you installed SunVTS*. If the test is not present in this directory, then it may only be available as a 32-bit test. For more information refer to the "32-Bit and 64-Bit Tests" section of the *SunVTS 5.1 Test Reference Manual* (816-5145-10).

Note – Errors returned by `zulutest` are nonspecific: It is not possible to determine which component caused a failure. In all error conditions, the field replaceable unit (FRU) is the entire Sun XVR-4000 graphics accelerator.

CHAPTER 7

Disk and Floppy Drives Test (disktest)

disktest verifies the functionality of hard drives and diskette drives using three subtests (see TABLE 7-1): Media, File System, and Asynchronous I/O. Most disk drives, such as SCSI disks, native or SCSI floppy disks, IPI, and so on, are supported. The type of drive being tested is displayed at the top of the Test Parameter Options dialog box.

The disktest Test Parameter Options dialog box shows all the partitions that are available for testing. The file System subtest can only be run if the selected partition is mounted (described below). The WriteRead option of the Media subtest is allowed only if a selected partition is *not* mounted.

disktest Test Requirements

By default, disktest does not mount any partitions. To have SunVTS pre-mount all mountable partitions, set the environment variable BYPASS_FS_PROBE to 0 (zero) before starting SunVTS. Pre-mounting can be disabled by unsetting BYPASS_FS_PROBE or changing it to a value other than 0 (zero).

The mount point used by disktest is the word *disktest* appended by the name of the disk partition. For example, if the disk partition name is /dev/dsk/c0t3d0s0, disktest mounts it as superuser under the name /disktest_c0t3d0s0.

Caution – If a power failure occurs OR if the disktest is terminated abruptly while the Media subtest is running in WriteRead mode, disk data may be corrupted.

Caution – Running the Media subtest on a disk partition in the WriteRead mode may cause data corruption if the same partition is being used by other programs. Only select this mode when the system is offline (not used by any other users or programs).

`disktest` tests the floppy drive regardless of whether the Volume Management software is running or not. The following mount point names are used:

- If the Volume Management software *is* running, `disktest` tests the disk drive with the mount point name in the `/etc/mnttab` file.
- If the Volume Management software *is not* running, `disktest` tests the disk drive with the device name `dev=/dev/diskette`. Do not edit the `/etc/vold.conf` file to change the diskette drives. Currently, the SunVTS software is hard-coded to use these path names as the default logic names.

Loading an option file (refer to the *SunVTS User's Guide* for option file details) that was created when `BYPASS_FS_PROBE` was set to `0` (zero) might not work if the `BYPASS_FS_PROBE` environment variable is no longer set to `0`. Testing may fail with the following error:

```
SUNWvts.disktest.8088 07/24/98 15:47:22 disktest c0t0d0 FATAL:
"Couldn't get file system information on /disktest_s0t0d0s0,
statvfs() system call failure error: No such file or directory.
```

This error is caused when SunVTS expects to use the predefined mount point names that are created when `BYPASS_FS_PROBE` is set to `0` (zero), but these mount points do not exist while `BYPASS_FS_PROBE` is not set to `0`.

To use option files with `disktest`, create two separate option files for the two different states of the `BYPASS_FS_PROBE` environment variable.

disktest Subtests

The following table describes the disktest subtests:

TABLE 7-1 disktest Subtests

Subtest	Description
Media subtest	The Media subtest verifies the disk media by allowing users to run disktest in different modes such as ReadOnly, ReadCompare, and WriteRead. The Media subtest treats the disk partition as one large chunk of contiguous data. In the WriteRead mode, all instances of disktest communicate through a shared memory service to ensure that they do not overlay the same disk area at the same time. This avoids data corruption. Each of the above three modes could run two different methods of disk testings. These are Synchronous I/O and Asynchronous I/O. SyncIO: Test reads and writes data using Read/Write system calls in a sequential fashion until the specified percentage of media is covered. AsyncIO: Test reads and writes data using aio library calls such as aioread(), aiowrite() until the specified percentage of media is covered. aiowait() is used to synchronize aio operations.
File System subtest	The File system subtest is used to verify the disk file system integrity. It exercises mounted disk partitions carrying the file system. By default, the test only runs on system-mounted partitions, it does not pre-mount any additional partitions. If you want SunVTS to pre-mount all of the unmounted partitions which have a file system, you have to set the environment variable BYPASS_FS_PROBE to '0' (zero). The test creates two temporary files of the size specified by File System File Size, writes the data patterns and compares the two files against each other.
Self subtest	This test is run as part of the Media subtest. You can not enable or disable this subtest. It is performed in Functional test mode only. This subtest instructs the disk to run its internal diagnostics. A failure in the Self subtest indicates a hardware problem with the actual device under test.
Write/Read device buffer subtest	This test is run as part of the Media subtest. You can not enable or disable this subtest. It is performed in Functional test mode only. This subtest verifies the Write/Read buffer for the device. This subtest uses the pattern specified for the Media subtest or the default pattern to write a defined number of iterations to the Write/Read buffer. A failure in the Write/Read buffer subtest indicates a problem in the upstream component and not with the actual test device.

`disktest` Test Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

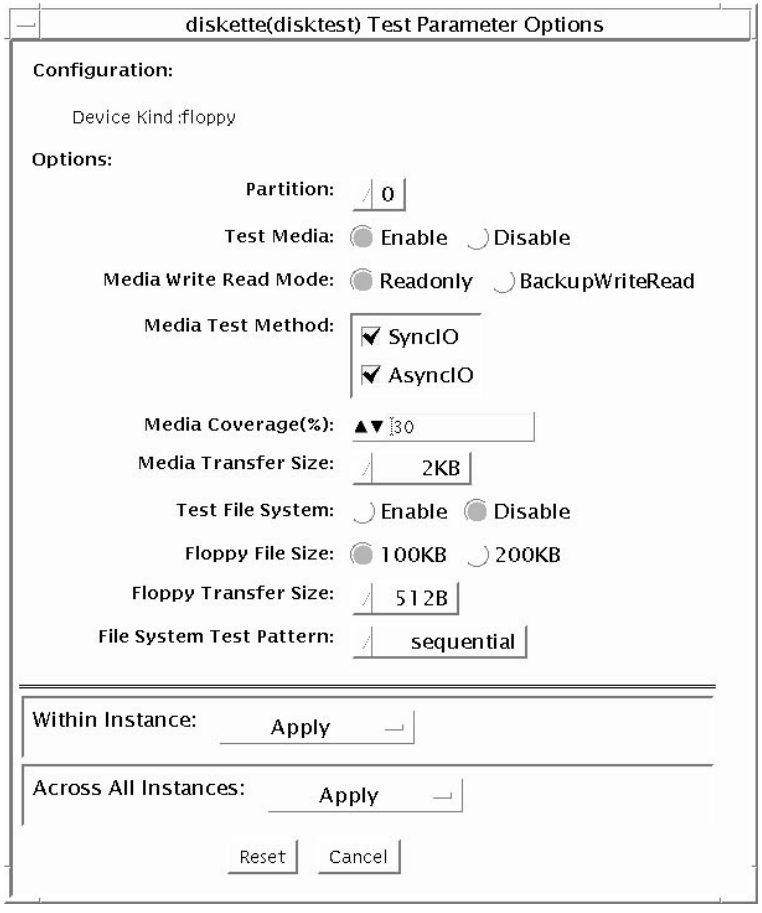

FIGURE 7-1 `disktest` Test Parameter Options Dialog Box

The following table describes the `disktest` option menu for different test modes.

TABLE 7-2 `disktest` Configurations and Options

`disktest` **Options**	**Description**
Partition	Displays the partition for the Media subtest. If a partition is mounted, its mount point is appended after the partition number, such as `1(/usr)`, where `1` is the partition number, and `"(/usr)"` is the mount point.
Test Media	Enables or disables the Media subtest.
Media Write Read Mode	Enables Read-Only or Compare after Read or Read after Write.
Media Test Method	Enables or disables the Media Test Methods (SyncIO and AsyncIO).
Media Coverage (% TB, GB, MB, KB, B)	Enables users to test all or part of a partition (in percentage or in any of TB, GB, MB, KB, B units)
Raw Test Pattern (P0 to P15)	Enables user to specify the write, read pattern. P0 – Low Frequency Pattern P1 – Low Transition Density Pattern P2 – High Transition Density Pattern P3 – Compliant Jitter Pattern P4 – Compliant Jitter: RPAT P5 – Compliant Jitter: CRPAT P6 – Compliant Jitter: JTPAT P7 – Compliant Jitter: CJTPAT P8 – Compliant Jitter: SPAT P9 – Compliant Jitter: CSPAT P10 – 8 Bit Cable Pattern P11 – 16 Bit Cable Pattern P12 – 8 Bit Xtalk Pattern P13 – 16 Bit Xtalk Pattern P14 – MFM Pattern P15 – Generic Test Patterns
Media Transfer Size	Displays the transfer size of the media subtest.
Test File System	Enables or disables the File System subtest.
File System File Size	Specifies the size for each of the two temporary files for File System testing.
File System Transfer Size	Displays the transfer size of the File System subtest.
File System Test Pattern	Test pattern of File System subtest.

TABLE 7-2 `disktest` Configurations and Options *(Continued)*

`disktest` **Options**	**Description**
Connection Test for Hard Disk	• Option Menu for hard disk partition—0 - 7 [default] • Test Media—[Enable](fixed to Enable) • Media Write Read Mode—[Read Only](fixed to Read Only) • Media Test Method-[SyncIO] (fixed to SyncIO) • Media Coverage(%)—1 • Media Transfer Size—[2KB] • Test File System—[Disable](fixed to Disable)
Online Mode for Hard Disk	• Partition—0 - 7 [default] • Test Media—[Enable] [Disable] • Test Mode—[Read-only~](fixed to Read-only) • Media Coverage(% TB, GB, MB, KB, B) • Media Transfer Size—[2KB]] • Test File System—[Disable~](fixed to Disable)
Functional Test for Hard Disk	• Partition—0 - 7 [default] • Test Media—[Enable] [Disable] • Media Write Read Mode—[Readonly] [CompareRead] [WriteRead] • Media Test method—[SyncIO] [AsyncIO] • Media Coverage(% TB, GB, MB, KB, B) Raw Test Pattern (P0 to P15) • Media Transfer Size—[2KB] [16KB] [32KB] [64KB] [128KB] [256KB] [512KB] • Test File System—[Enable] [Disable] • File System File Size—[512KB] [2MB] [8MB] [20MB] [100MB] [200MB] • File System Transfer Size—[512B] [1024B] [10KB] [40KB] [80KB] • File System Test Pattern—[sequential] [0x00000000] [0xffffffff] [0x5aa55aa5] [0xdb6db6db] [random]
Functional Test for Floppy Disk	• (under Other-Devices group)—partition: 0 - 7 [default] • Test Media—[Enable]- [Disable] • Media Write Read Mode—[Read-only] [BackupWriteRead] • Media Test Method—[SyncIO] [AsyncIO] • Media Coverage(% TB, GB, MB, KB, B) Raw Test Pattern (P0 to P15) • Media Transfer Size—[2KB] [10KB] [20KB] • Test File System—[Enable] [Disable] • Floppy File Size— [100KB] [200KB] • Floppy Transfer Size—[512B] [1024B] [10KB] • File System Test Pattern—[sequential] [0x00000000] [0xffffffff] [0x5aa55aa5] [0xdb6db6db] [random]

disktest Test Modes

TABLE 7-3 disktest Supported Test Modes

Test Mode	Description
Connection	Only one instance of disktest (which monitors UNIX error messages) is allowed for each disk device. disktest displays messages and reports errors. The test also opens the hard disk, checks the disk configuration, reads a few blocks, and then closes the hard disk. No File System subtest is run. No Write option is available in Connection test mode.
Functional	More than one instance of disktest is allowed for one disk device. The File System subtest, Media subtests, and floppy test can be run in Functional test mode. In Functional mode, disktest performs two additional subtests (Self subtest and Write/Read device buffer subtest) for enclosures. These two additional subtests help in isolating the errors and are completed before disktest continues with the Media subtest or File System subtest. In Functional test mode, disktest also monitors enclosures by checking for errors in the Read link status counters and issues a warning if any errors are detected.

disktest Command-Line Syntax

/opt/SUNWvts/bin/disktest *standard_arguments* **-o partition=**<*0-7*> *"(mount_point)"***,rawsub=***E* | *D***,rawrw=***Readonly* | *CompareRead* | *WriteRead***, rawiosize=***n***,rawcover=***n***,method=***AsyncIO+SyncIO***,fssub=***E* | *D***,fssize=***n***, fsiosize=***n***,fspattern=***data_pattern***,**dev=device_name**,rawpattern=**<*P* | *0x*<*8 number pattern*>*{0* | *1* | *2* | *3* | *4* | *5* | *6* | *8* | *9* | *10* | *11* | *12* | *13* | *14* | *15}*

TABLE 7-4 disktest Command-Line Syntax

Argument	Description
partition=<*0-7*>**"(***mount_point***)"**	Specifies the partition number as follows: • *n*—is the partition number (slice number), usually *0-7* • *mount_point*—is the mount point for the mounted partition that you plan to test For example: **partition=6"(/export)"**
rawsub= *E(nable)* | *D(isable)*	Enables or disables the Media subtest. For example: **rawsub=** *Enable*
rawrw= *Readonly* | *CompareRead* | *WriteRead*	Specifies the Media subtest Read, Compare, and Write mode: • Read only • Read twice, Compare (works only with SyncIO method) • Write, Read, Compare, Restore For example: **rawrw=***ReadOnly*
method=*AsyncIO+SyncIO*	Specifies the Media access method. You can choose to use either or both methods . If you use both access methods together, you must insert a '+' between the two: AsyncIO: Runs the asynchronous i/o test, using the async read/write feature of the Solaris disk driver SyncIO: Runs the synchronous i/o test. For example: **method=***AsyncIO*
rawcover=n	Specifies media coverage from 0-100 (percentage) of the partition. Media Coverage can also be specified in units: TB, GB, MB, KB and B. For example: **rawcover=***40* OR **rawcover=***4GB*

TABLE 7-4 `disktest` Command-Line Syntax *(Continued)*

Argument	Description
`rawiosize=`n	Specifies the media size to transfer. The block size can be specified in kilobytes. For example: 2K,...512K. For example: **`rawiosize=`**9
`rawpattern=`*<P | 0x<8 number pattern>{0 | 1 | 2 | 3 | 4 | 5 | 6 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15}*	`rawpattern` could be specified as a pre-defined pattern set, *P(0-15)*, or an 8 digit pattern could be specified as: *0xaa55aa55+0xff00ff00+0x*. The following is a description of the supported pre-defined patterns: P0 – Low Frequency Pattern P1 – Low Transition Density Pattern P2 – High Transition Density Pattern P3 – Compliant Jitter Pattern P4 – Compliant Jitter: RPAT P5 – Compliant Jitter: CRPAT P6 – Compliant Jitter: JTPAT P7 – Compliant Jitter: CJTPAT P8 – Compliant Jitter: SPAT P9 – Compliant Jitter: CSPAT P10 – 8 Bit Cable Pattern P11 – 16 Bit Cable Pattern P12 – 8 Bit Xtalk Pattern P13 – 16 Bit Xtalk Pattern P14 – MFM Pattern P15 – Generic Test Patterns For example: **`rawpattern=`***P1*
`fssub=`*E(nable) | D(isable)*	Enables or disables the File System subtest.
`fspattern=`*data_pattern*	Specifies the file system data pattern as sequential or random. *{seq(uential) | 0x0(0000000) | 0xf(ffffffff) | 0xa (5a5a5a5) | 0x5(a5a5a5a) | ran(dom) | 0xd(b6db6db)}* For example: **`fspattern=`***data_pattern*

TABLE 7-4 `disktest` Command-Line Syntax *(Continued)*

Argument	Description
`fssize=`*n*	Indicates the file system subtest size in kilobytes or megabytes: • `K|k|KB|kb` – kilobytes • `M|m|MB|mb` – megabytes `512KB|2MB| 8MB| 20MB|100MB|200MB`
`fsiosize=`*n*	Indicates the size of the file system subtest I/O transfer in bytes or kilobytes: • B|b – bytes • `K|k|KB|kb` – kilobytes `512B|1024B|10KB|40KB|80KB`
`dev=`*device_name*	Specifies the name of the disk to be tested, such as c0t3d0.

The following example shows how to run `disktest` on a partition "0" (which is mounted under "/") for the disk device `c0t0d0`. The media subtest is enabled in ReadOnly mode using SyncIO method. The coverage specified is 30% with 512KB transfer size. The File System subtest is disabled.

```
# /opt/SUNWvts/bin/disktest -f -o partition=0"(/)", rawsub=Enable,
rawrw=ReadOnly, method=SyncIO, rawcover=30, rawiosize=512KB,
fssub=Disable, dev=c0t0d0
```

Note – 64-bit tests are located in the `sparcv9` subdirectory `/opt/SUNWvts/bin/sparcv9/`*testname, or the relative path to which you installed SunVTS*. If a test is not present in this directory, then it may only be available as a 32-bit test. For more information refer to the "32-Bit and 64-Bit Tests" section of the *SunVTS 5.1 Test Reference Manual* (816-5145-10).

CHAPTER 8

System Test (`systest`)

The `systest` checks the overall functionality of a Sun system by exercising the CPU, I/O, and Memory channels simultaneously. The test ensures the concurrency of the different channels by the use of Solaris threads. The test aims at stimulating failures that might be caused due to the interaction of the various different hardware modules in the system. It is very stressful on the CPU, and stresses the parallel computational capability of a multiprocessor system.

`systest` Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

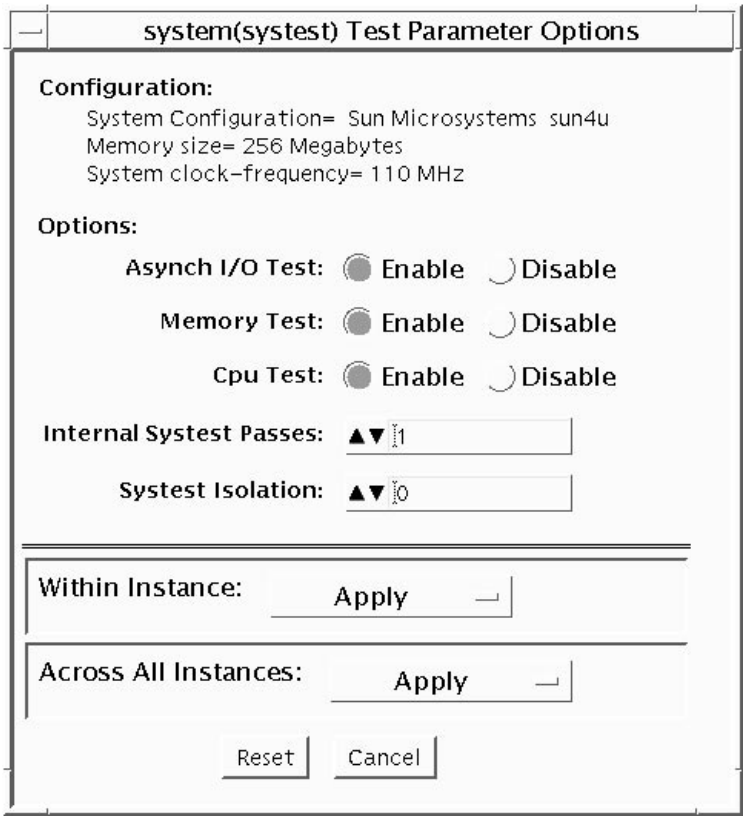

FIGURE 8-1 `systest` Test Parameter Options Dialog Box

Caution – Use discretion when defining the syspass parameter. One linpack pass (syspass=1) takes approximately 40 minutes on a server with 12 UltraSPARC™ III processors. If the syspass value is set to a high value, it also increases the probability of detecting residual errors.

Caution – Use strong discretion when defining the System Isolation (sysiso) parameter. BE AWARE THAT sysiso MAY ONLINE / OFFLINE CPUs IN THE SYSTEM. DO NOT USE sysiso ON PRODUCTION SERVERS. If you choose CPUs (sysiso=2) Isolation, the run time may be much higher than for board(s) (sysiso=1) Isolation. The total run time for Isolation can not be precisely estimated. If a residual error is found in the initial evaluation phase, the Isolation functionality will online / offline CPUs in order to detect the defective boards and CPUs in the system.

Note – Users are advised to not use the Processor Affinity option for this test. Doing so reduces the effectiveness of the test.

TABLE 8-1 `systest` Options

`systest` **Options**	**Description**
Asynch I/O Test	Enables or disables the Asynch I/O subtest. The default is enable.
Memory Test	Enables or disables the Memory subtest. The default is enable.
CPU Test	Enables or disables the CPU/FPU subtests. The default is enable.
Internal System Passes	Defines the number of internal linpack passes. A set of boards and CPUs will be declared "GOOD" after "`syspass`" number of passes. The default is 1.
System Isolation	Defines the type of Isolation that `systest` needs to perform if a residual error is found in the initial evaluation phase. 0 = No Isolation (default) 1 = Board(s) Isolation only 2 = Board(s) and CPUs Isolation

The default values are recommended for an initial evaluation of the system.

systest Test Modes

TABLE 8-2 systest Supported Test Modes

Test Mode	Description
Exclusive	Performs only the systest (full test).

systest Command-Line Syntax

/opt/SUNWvts/bin/systest *standard_arguments* **-o -io=***Enable* | *Disable* **-mem=***Enable* | *Disable*, **-cpu=***Enable* | *Disable*, **-dev=**system, **-syspass=***1,2000*, **-sysiso=***0* | *1* | *2*

TABLE 8-3 systest Command Line Syntax

Argument	Description
io=*Enable*	*Disable*	Enables or disables the Asynch I/O subtest.
mem=*Enable*	*Disable*	Enables or disables the Memory subtest.
cpu=*Enable*	*Disable*	Enables or disables the CPU/FPU subtests.
dev=*system*	Specifies the pseudo device name.
syspass=*1,2000*	Defines the number of internal linpack passes. A set of boards and CPUs will be declared "GOOD" after "syspass" number of passes. The default is 1.
sysiso=*0*	*1*	*2*	Defines the type of Isolation that systest needs to perform if a residual error is found in the initial evaluation phase. 0 = No Isolation 1 = Board(s) Isolation only 2 = Board(s) and CPUs Isolation

Note – 64-bit tests are located in the sparcv9 subdirectory: /opt/SUNWvts/bin/sparcv9/*testname, or the relative path to which you installed SunVTS*. If a test is not present in this directory, then it may only be available as a 32-bit test. For more information refer to the "32-Bit and 64-Bit Tests" section of the *SunVTS 5.1 Test Reference Manual* (816-5145-10).

Recommended Option Selection

The default values are recommended for an initial evaluation of the system.

Command-Line Examples

The following examples assume the user wants to execute `systest` from the command-line with verbose enabled.

Example 1:

```
# ./systest  -xv
```

The above example invokes the following:

- `systest` with default parameter values
- I/O, MEM, and CPU subtests
- One internal pass of linpack and no Isolation

Example 2:

```
# ./systest  -xv -o io=Disable,mem=Enable,cpu=Enable,dev=system
```

The above example invokes the following:

- `systest` without the I/O subtest
- MEM and CPU subtests
- One internal pass of linpack and no Isolation

Caution – Do not perform the following systest examples (3 and 4) on production servers because systest may online / offline CPUs.

Example 3:

```
# ./systest  -xv -o syspass=15,sysiso=1
```

The above example invokes the following:

- I/O, MEM, and CPU subtests
- Declares a set of boards free from residual errors after 15 internal passes of the linpack algorithm

- If an error is found, `systest` will perform boards isolation

Example 4:

```
# ./systest  -xv -o syspass=10,sysiso=2
```

The above example invokes the following:

- I/O, MEM, and CPU subtests
- Declares a set of boards AND CPUs free from residual errors after 10 internal passes of the linpack algorithm
- If an error is found, `systest` will perform boards AND CPUs isolation

CHAPTER 9

Virtual Memory Test (vmemtest)

The vmemtest checks virtual memory; that is, it tests the combination of physical memory and the swap partitions of the disk(s).

Note – This test may not stop immediately after being disabled.

This test uses the Solaris valloc (page aligned) system call to allocate, write, read, and compare virtual memory. These operations normally cause heavy paging activity on the system and simulate a stressful environment for the operating system. This test also detects ECC parity errors, memory read errors, and addressing problems, and displays the corresponding virtual memory addresses on failure.

Note – Do not run the vmemtest with fwcamtest at the same time on any Sun Blade™ system. This will cause the test to fail.

vmemtest Swap Space Requirements

Running this test places a significant burden on the operating system, since it uses the majority of swap space available for testing. You should use the vmemtest swap space reserve option when non-SunVTS test processes are started after SunVTS testing has started. See "Swap Space Requirements" in the *SunVTS User's Guide* for a complete discussion of swap space requirements.

`vmemtest` Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

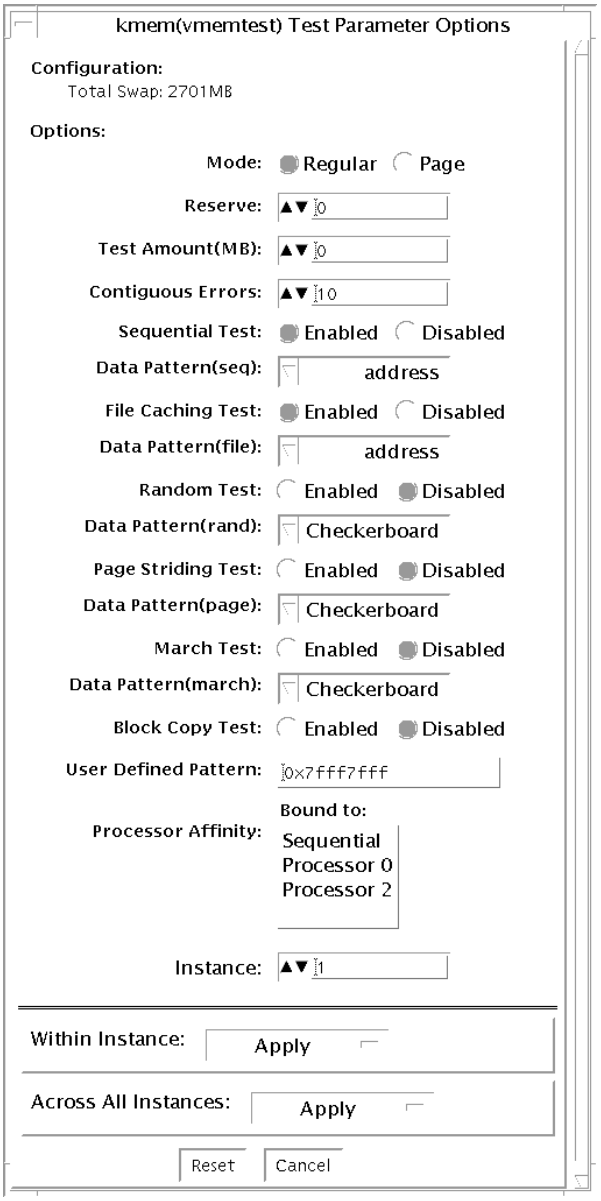

FIGURE 9-1 `vmemtest` Test Parameter Options Dialog Box

TABLE 9-1 `vmemtest` Options

vmemtest Options	Description
Mode	Two modes are available: • Regular mode tests the specified amount of memory as one chunk and passed as the size arugment to the different test algorithm functions (subtests). • Page mode tests assign virtual memory one page at a time. Each page is mapped to the temporary file /tmp/vmem.page and is then paged out to storage once test data is written. Next, the temporary page is paged back into memory for a read and compare. vmemtest runs in Regular mode as default setting
Reserve	The Reserve option specifies the amount of memory to reserve from being tested by vmemtest. The test ensures this amount of memory is left free on the system while evaluating the size of memory for testing. If specified value of reserve is Zero, test will use a default value evaluated based on the available free swap space for the instance. Trying to reserve more memory than available free memory by this instance will cause the test to fail.
Test Amount	An amount can be specified to test the virtual memory, instead of the default.The default value is 0, which means the default memory size is evaluated within the test. It is desirable to the user to know the memory configuration details on the target system while choosing Non default setting for "amount" option. If negetive values are specified, test will assume default setting while it runs. The actual size of memory tested by the instance is always evaluated with reference to the available free swap space on the system.
`vmemtest` Configuration	The amount of memory listed in the Configuration field is equivalent to the sum of the used and available swap space amounts returned by the swap -s command. It indicates the amount of virtual memory found, rounded up to the nearest Kbyte.
Contiguous Errors	Specifies the max. number of contigious memory errors, which will be considered and counted as one non contigious error. The default value is 10.

TABLE 9-1 `vmemtest` Options

vmemtest Options	Description
Test Method	• vmemtest runs Sequential subtest and FileCache subtest by default. • Sequential subtest : The whole memory is tested from the beginning address to the end address in a sequence. • Address Random subtest: Randomly selects memory addresses with in the specified range to test. Page Striding subtest: Non-contiguous memory test, implemented sequentially and non-sequentially. Sequential striding: tests from the first page to the last page, withn a specified test range. Only one word is tested per page. Non-sequential striding: tests randomly from first to last page, within a specified memory range. Goes back and forth testing one word per page until all pages are tested. Block Copy test: Writes and reads data between two memory blocks. Each memory block is half the memory to be tested. File Caching test: Aimed at improving performance through the use of file caching in the Solaris kernel. This test is useful for large memory configurations. This test takes 30 to 70% less time than the Sequential test method.
Predefined Pattern	• Select one of the following patterns to use for the test: • Address--uses the virtual addresses of the tested memory locations. • walk_1--uses a pattern that starts with 0x80000000 through 0x11111111 • walk_0--uses a pattern that starts with 0x7fffffff through 0x00000000 • 0x00000000--uses all ones and zeros for testing • 0x5aa55aa5--uses 0x5aa55aa5 pattern • 0xdb6db6db--uses 0xdb6db6db pattern • Checkerboard--uses 0xaaaaaaaa patterns. • UserDefined--uses the pattern that is specified in the User Defined Pattern area (see below).
User Defined Pattern	Only used if the Predefined Pattern is set to UserDefined. The pattern specified should be in the form of an 8-digit, hexadecimal number such as 0x2a341234. Default setting is up= 0x7ff77ff7
Instance	Specifies how many copies of the vmemtest test to run.

vmemtest Test Modes

TABLE 9-2 vmemtest Supported Test Modes

Test Mode	Description
Functional	Runs the full set of tests.

In Functional test mode, vmemtest writes a pattern to an amount of virtual memory specified by the user. Then the data is read back and compared. If there is a miscompare, the data is read again and compared. Whenever there is a miscompare, the virtual address is reported. When there is a miscompare on recomparison, an attempt is made to convert the virtual address to the physical address if the SunVTS diagnostic driver is installed.

vmemtest Command-Line Syntax

/opt/SUNWvts/bin/vmemtest *standard_arguments* **-o mode=***type*, **reserve=** *n*, **amount=***n*, **type1=***Enable* | *Disable*, **pp1=***pattern*, **type2=***Enable* | *Disable*, **pp2=***pattern*, **type3=***n*, **pp3=***patten*, **type4=***Enable* | *Disable*, **pp4=***pattern*, **type5=***Enable* | *Disable*, **pp5=***pattern*, **type6=***Enable* | *Disable*, **up=***hex_pattern*

TABLE 9-3 vmemtest Command-Line Syntax

Argument	Explanation
mode= Page|Regular	Specifies which mode of the vmemtest to run. Choose: • Page—tells the write/read memory test to proceed one system memory page at a time. • Regular—uses the valloc option to allocate the entire assigned memory, which is read and compared one long word at a time.
reserve=*n*	Specifies the amount of MB of virtual memory to reserve.
amount=*n*	Specifies the number of MB of memory to be tested instead of the default.
cerr=*n*	Specifies the maximum number of contiguous errors to be counted as one non contiguous error.

TABLE 9-3 `vmemtest` Command-Line Syntax

Argument	*(Continued)*Explanation
type1=*value* **pp1=***pattern*	`type1` is sequential test. The value is Enabled or Disabled; the default is Enabled. The default for the `pp1` pattern is `address`; select the `pp1` pattern from: **address**,**walk_0**,**walk_1**,**Checkerboard**, **0x00000000**,**0xffffffff**,**0x5aa55aa5**, **0xdb6db6db**,**random**,*UserDefined*
type2=*value* **pp1=***pattern*	`type2` is File cache test. The value is Enabled or Disabled; the default is Enabled. The default for the `pp1` pattern is `address`; select the `pp1` pattern from: **address**,**walk_0**,**walk_1**,**Checkerboard**, **0x00000000**,**0xffffffff**,**0x5aa55aa5**,**0xdb6db6db**, **random**,*UserDefined*
type3=*value* **pp3=***pattern*	`type3` is Random address test. The value is Enabled or Disabled; the default is Disabled. The default of the `pp3` pattern is `checkerboard`; select the `pp3` pattern from: **Checkerboard**,**0x00000000**,**0xffffffff**, **0x5aa55aa5**,**0xdb6db6db**,*UserDefined*
type4=*value* **pp4=***pattern*	`type4` is `page_striding` test. The value is Enabled or Disabled; the default is Disabled. The default of the `pp4` pattern is `checkerboard`; select the `pp4` pattern from: **Checkerboard**,**0x00000000**,**0xffffffff**, **0x5aa55aa5**,**0xdb6db6db**,*UserDefined*
type5=*value* **pp5=***pattern*	`type5` is `march_c` test. The value is Enabled or Disabled; the default is Disabled. The default for the `pp5` pattern is `checkerboard`; select the `pp5` pattern from: **Checkerboard**,**0x00000000**,**0xffffffff**, **0x5aa55aa5**,**0xdb6db6db**,*UserDefined*
type6=*value*	`type6` is `Block_Copy` test. The value is Enabled or Disabled; the default is Disabled. Note – The `Block_Copy` subtest uses its own set of the data patterns predefined in the test. It does not require any user specified data patterns for testing.
up=*hex_address*	Only used if the `pp` argument is set to `UserDefined`. The pattern specified should be in the form of a 8-digit, hexadecimal number such as `0x2a341234`.

Note – 64-bit tests are located in the `sparcv9` subdirectory: `/opt/SUNWvts/bin/sparcv9/`*testname, or the relative path to which you installed SunVTS*. If a test is not present in this directory, then it may only be available as a 32-bit test. For more information refer to the "32-Bit and 64-Bit Tests" section of the *SunVTS 5.1 Test Reference Manual* (816-5145-10).

CHAPTER 10

Blade Support Chip Test (bsctest)

The bsctest exercises the Blade Support Chip and supporting hardware used in Sun Fire™ B100 blade systems. This includes the Open Boot Prom (OBP) and Time of Day (ToD) Prom chips.

Caution – If the LED subtest is selected, please be aware that LEDs on the blade will change. They will return to their correct state when the test is completed.

bsctest Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

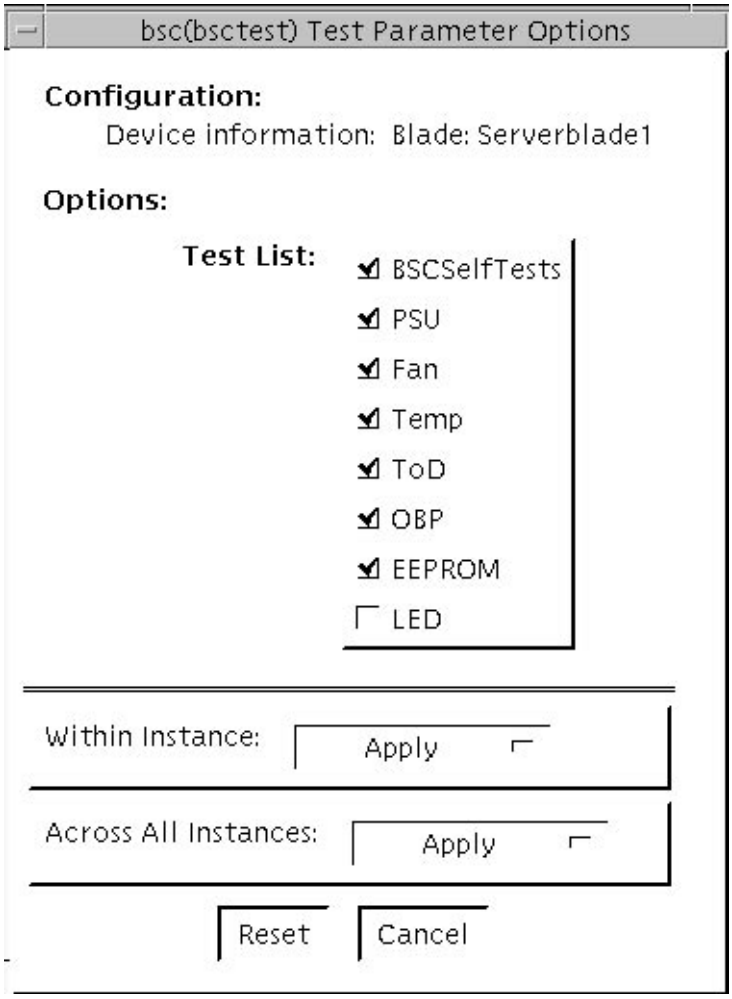

FIGURE 10-1 `bsctest` Test Parameter Options Dialog Box

TABLE 10-1 `bsctest` Options

`bsctest` **Options**	**Description**
BSCSelfTests	Calls on the BSC to execute its built-in self tests.
PSU	Performs read-only checks of Power Supply status.
Fan	Performs read-only checks of Fan status.
Temp	Performs read-only checks of Temperature Monitor status.
ToD	Performs read-only checks of Time of Day chip.

TABLE 10-1 `bsctest` Options *(Continued)*

`bsctest` **Options**	**Description**
OBP	Performs read-only checks of platform specific Open Boot properties.
EEPROM	Performs read-only check of EEPROM.
LED	Performs read-only check of Service Required LED status and performs a test in which all three LEDs (Power, Service Required, and Ready to Remove) are flashed simultaneously at 4Hz and then returned to their original state.

`bsctest` Test Modes

TABLE 10-2 `bsctest` Supported Test Modes

Test Mode	**Description**
Connection	Opens the BSC, OBP, and ToD devices.
Functional	Performs all tests with the LED testing off by default.
Online	Performs all tests except BSCSelfTests and LED *Flashing* test.

`bsctest` Command-Line Syntax

`/opt/SUNWvts/bin/bsctest` *standard_arguments* **`[-o dev`**=*device_name* **`test`**=<*test_list*>**`]]`**

TABLE 10-3 `bsctest` Command-Line Syntax

Argument	**Description**
`dev=`*device_name*	*device_name* is the device to be tested, for example, `bsc`
`test`=*test_list*	*testlist* is the list of subtests, for example: `BSCSelfTests`, `PSU`, `Fan`, `Temp`, `ToD`, `OBP`, `EEPROM`, `LED`

Note – 64-bit tests are located in the `sparcv9` subdirectory: `/opt/SUNWvts/bin/sparcv9/`*testname, or the relative path to which you installed SunVTS*. If the test is not present in this directory, then it may only be available as a 32-bit test. For more information refer to the "32-Bit and 64-Bit Tests" section of the *SunVTS 5.1 Test Reference Manual* (816-5145-10).

CHAPTER 11

Environmental Test (env6test)

env6test exercises and validates environmental subsytems. This test contains subtests to exercise a system's fans, keyswitch, LEDs, power supplies, and temperature sensors.

This test is not scalable.

env6test Test Requirements

In Functional test mode, if two or more of the following tests: ssptest, i2c2test, or env6test are selected under the "Other Devices" category, the group concurrency for this category should be set to 1. This setting will prevent any potential test failures due to the limitations in the Host to ALOM (Advanced Lights-Out Manager) interface.

env6test Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

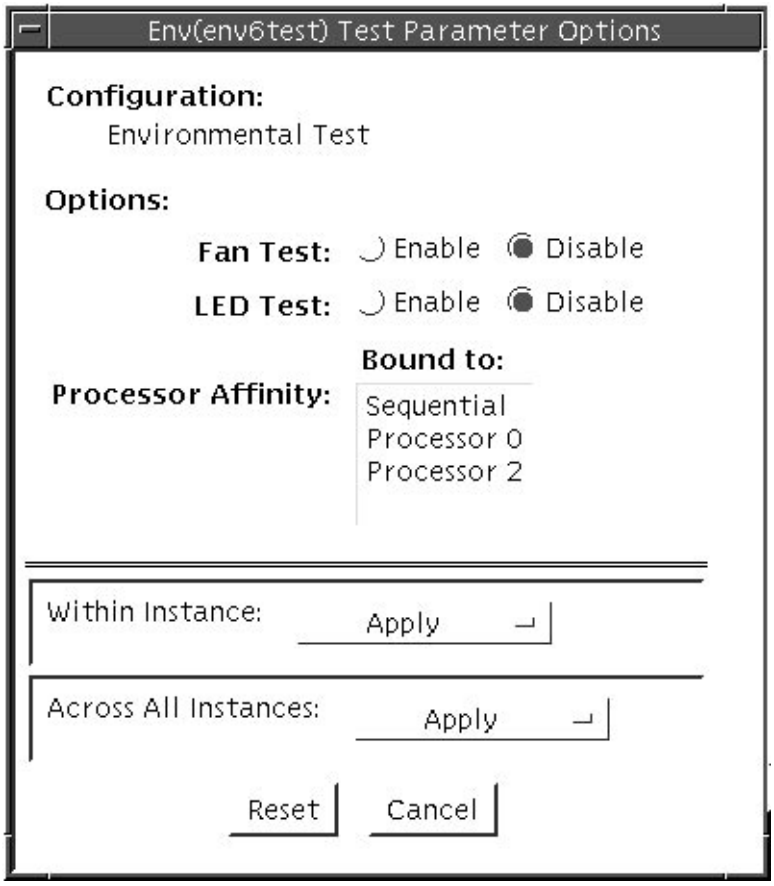

FIGURE 11-1 `env6test` Parameter Options Dialog Box

TABLE 11-1 `env6test` Options

`env6test` **Options**	**Description**
Fan Test	Checks the status, tolerance, and speed of the system's fans. Default is Disable.
LED Test	Checks overall status of system's LEDs by switching them ON and OFF. Default is Disable.

env6test Test Modes

TABLE 11-2 env6test Supported Test Modes

Test Mode	Description
Connection	Reports current state of devices.
Functional	Performs all tests including the Fan and LED subtests if they are enabled.
Exclusive	Performs all tests including the Fan and LED subtests if they are enabled.

env6test Command-Line Syntax

/opt/SUNWvts/bin/env6test *standard_arguments*
-o dev=*raw_device_name***,led=***Enable* | *Disable***,fan=***Enable* | *Disable*

TABLE 11-3 env6test Command-Line Syntax

Argument	Description
dev=*raw_device_name*	Specifies the name of the raw device to test. Default is /dev/env
led=*Enable*	*Disable*	Enables or disables the LEDS subtest. Default is Disable.
fan=*Enable*	*Disable*	Enables or disables the Fans subtest. Default is Disable.

Note – 64-bit tests are located in the sparcv9 subdirectory: /opt/SUNWvts/bin/sparcv9/*testname*, or the relative path to which you installed SunVTS. If a test is not present in this directory, then it may only be available as a 32-bit test. For more information refer to the "32-Bit and 64-Bit Tests" section of the *SunVTS 5.1 Test Reference Manual* (816-5145-10).

CHAPTER 12

I2C Inter-Integrated Circuit Test (`i2c2test`)

The `i2c2test` is designed to verify the proper placement, operation, and data integrity on the various I2C devices.

This test is not scalable.

`i2c2test` Test Requirements

In Functional test mode, if two or more of the following tests: `sspteste`, `i2c2test`, or `env6test` are selected under the "Other Devices" category, the group concurrency for this category should be set to 1. This setting will prevent any potential test failures due to the limitations in the Host to ALOM (Advanced Lights-Out Manager) interface.

`i2c2test` Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

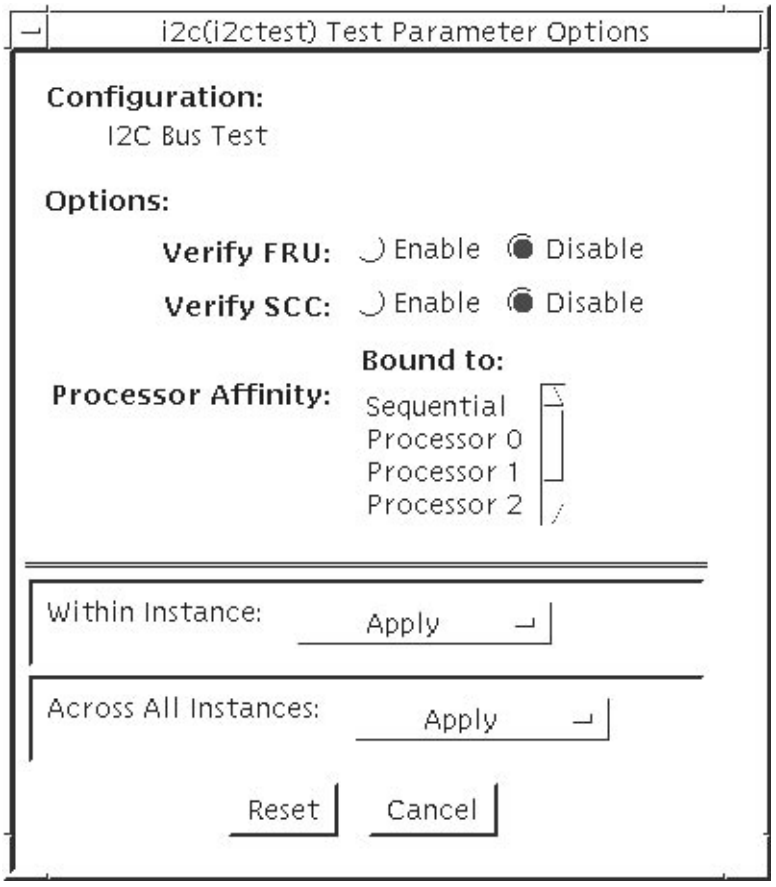

FIGURE 12-1 `i2c2test` Test Parameter Options Dialog Box

TABLE 12-1 `i2c2test` Options

`i2c2test` **Options**	**Description**
Verify FRU	Verifies the status of the FRU. Default is Disable.
Verify SCC	Verifies the status of the SCC. Default is Disable.
Processor Affinity	Specifies the processors to be tested in sequential order.

i2c2test Test Modes

TABLE 12-2 i2c2test Supported Test Modes

Test Mode	Description
Connection	Performs a test to verify connection to all I2C devices.
Functional	Performs a test to verify connection to all I2C devices, and also performs a test to verify that the fru and scc content is consistent with the user's selection.
Exclusive	Performs a test to verify connection to all I2C devices, and also performs a test to verify that the fru and scc content is consistent with the user's selection.

i2c2test Command-Line Syntax

/opt/SUNWvts/bin/i2c2test *standard_arguments*
-o dev=*raw_device_name***,chkfru**=*Enable* | *Disable***,chkscc**=*Enable* | *Disable*

TABLE 12-3 i2c2test Command-Line Syntax

Argument	Description
dev=*raw_device_name*	Specifies the name of the raw device to test.
chkfru=*Enable*	*Disable*	Verifies the status of the FRU. Default is Disable.
chkscc=*Enable*	*Disable*	Verifies the status of the SCC. Default is Disable.

Note – 64-bit tests are located in the sparcv9 subdirectory: /opt/SUNWvts/bin/sparcv9/*testname*. If a test is not present in this directory, then it may only be available as a 32-bit test. For more information refer to the "32-Bit and 64-Bit Tests" section of the *SunVTS 5.1 Test Reference Manual* (816-5145-10).

CHAPTER 13

Integer Unit Test (`iutest`)

The Integer Unit Test (iutest) tests the resident integer unit in Sun SPARC CPUs. It exercises all of the register windows present in the Integer Unit of the CPU. The successful completion of the test implies that all of the register windows are functioning properly and failure implies a faulty register.

`iutest` Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

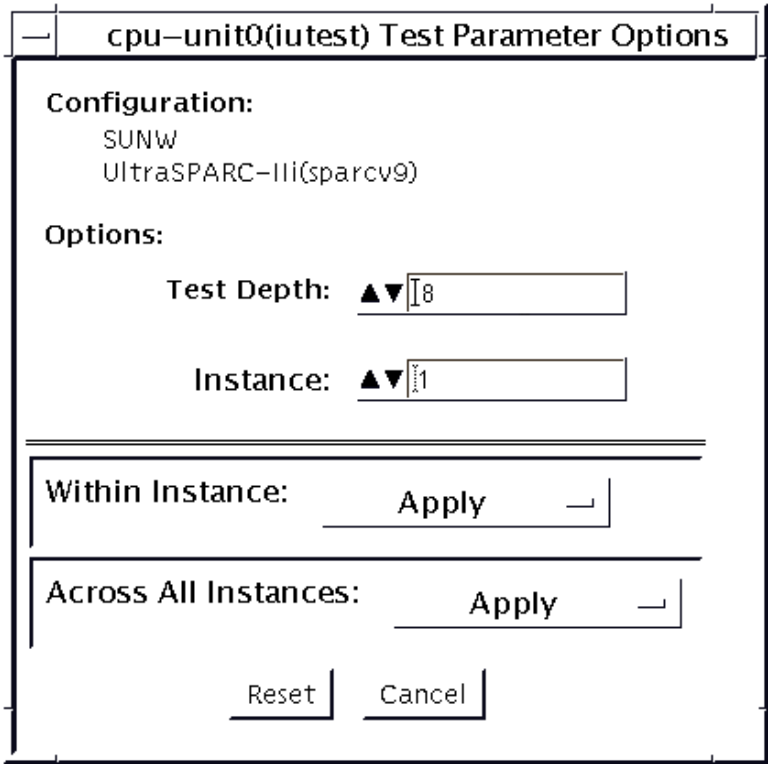

FIGURE 13-1 `iutest` Test Parameter Options Dialog Box

For the test options in the `iutest` Test Parameter Options dialog box, Test Depth is the only required option. Test Depth corresponds to the number of times that *all the register windows* are getting tested. The default, maximum and minimum values of the Test Depth are 8, 64, and 1 respectively.

iutest Test Modes

TABLE 13-1 iutest Supported Test Modes

Test Mode	Description
Connection	Displays the type of CPU implementation (for example, sparcv7 or sparcv9, etc.), the operating frequency, and CPU status (online, offline, etc.).
Functional (Offline)	Verifies all of the register windows and returns the appropriate error message if there is a faulty register. Otherwise, displays a successful test message.

iutest Command-Line Syntax

/opt/SUNWvts/bin/iutest *standard_arguments* **-o depth=***val*,**dev=cpu-unit***N*

In the iutest command-line syntax, *val* is the value of the Test_Depth parameter option as described in the preceding iutest options section. *N* is the CPU unit number (0,1,2, etc.). The test behavior is unpredictable if options other than those described in this section are entered.

Note – 64-bit tests are located in the sparcv9 subdirectory: /opt/SUNWvts/bin/sparcv9/*testname, or the relative path to which you installed SunVTS*. If a test is not present in this directory, then it may only be available as a 32-bit test. For more information refer to the "32-Bit and 64-Bit Tests" section of the *SunVTS 5.1 Test Reference Manual* (816-5145-10).

CHAPTER 14

System Service Processor Test (ssptest)

The ssptest exercises the Remote System Control (RSC) feature, which is integrated on the Sun Enterprise 250 as well as the next-generation RSC 2.0 plug-in card introduced with the Sun Fire 280R line, and Advanced Lights-Out Management (ALOM) feature. The ALOM feature is integrated into the Sun Fire V210 and Sun Fire V240.

The RSC or ALOM provides secure remote access for system monitoring, firmware updates, and failure recovery. The RSC or ALOM communicates with the host through two internal serial lines, the I2C bus, and reset lines.

The RSC 1.0 hardware consists of the controller, flash, SEEPROM, 10MB Ethernet port, and an external console serial port.

The RSC 2.0 plug-in card hardware consists of the controller, flash, SEEPROM, 10MB Ethernet port, FRUSEEPROM, Time of Day (ToD) device, internal PCMCIA modem card, and battery backup.

The ALOM hardware consists of a Motorola MPC850 processor, flash, SEEPROM, 10MB/100MB Ethernet Port, Time of Day (ToD) device, Serial Transceiver, and battery backup.

ssptest is not scalable.

ssptest Requirements

In Functional test mode, if two or more of the following tests: `ssptest`, `i2c2test`, or `env6test` are selected under the "Other Devices" category, the group concurrency for this category should be set to 1. This setting will prevent any potential test failures due to the limitations in the Host to ALOM (Advanced Lights-Out Manager) interface.

ssptest Subtests

The `ssptest` will present different subtests and options based on which type of hardware (RSC or ALOM) and which version of RSC hardware (1.0 or 2.0) it is testing.

The subtests common to RSC 1.0, RSC 2.0, and ALOM include:

TABLE 14-1 Subtests for Both RSC 1.0, RSC 2.0, and ALOM

Subtest	Description
Ethernet	Allows for internal loopback testing, on the Ethernet device with user specified data, size, and number of packets.
	Allows for external loopback testing with user-specified data, size, and number of packets. This requires a connection to a 10MB hub or switch for RSC 1.0, or a passive loopback connector for RSC 2.0, and ALOM.
	Allows for a ping to be sent to a specified host and checks the response.
Flash CRC	Performs a checksum test on the flash device.
SEEPROM CRC	Performs a checksum test on the SEEPROM device.
Serial	Allows internal loopback testing with user-specified data and size on the two internal serial ports.
	Allows for internal and/or external testing on the external `ttyu` port. The external test requires a passive loopback connector.

`ssptest` also presents the following subtests when running on the RSC 2.0 hardware:

TABLE 14-2 Subtests for RSC 2.0 Only

Subtest	Description
FRU SEEPROM CRC	Performs a checksum test on the SEEPROM device.
I2C	Tests the i2c bus connection between the host and the RSC.
ToD	Performs multiple reads to the ToD device and verifies that the time is incrementing.
Modem	Verifies that the modem is installed. Displays the manufacture information, in Verbose mode. Performs AT inquiry commands.

`ssptest` presents the following subtests when running on the ALOM hardware:

TABLE 14-3 Subtests for ALOM Only

Subtest	Description
I2C	Tests the i2c bus connection between the host and the ALOM.
ToD	Performs multiple reads to the ToD device and verifies that the time is incrementing.

The subtests call test modlets that are written in the native Real Time Operating System (RTOS) that resides in the RSC firmware. The `ssptest` subtests execute the test modlets, pass parameters, and retrieve results from the RSC or ALOM using a test protocol on the host to RSC or ALOM internal serial lines.

`ssptest` Options

To reach the dialog box below, right-click on the test name in the System Map and select Test Parameter Options. If you do not see this test in the System Map, you might need to expand the collapsed groups, or your system may not include the device appropriate to this test. Refer to the *SunVTS User's Guide* for more details.

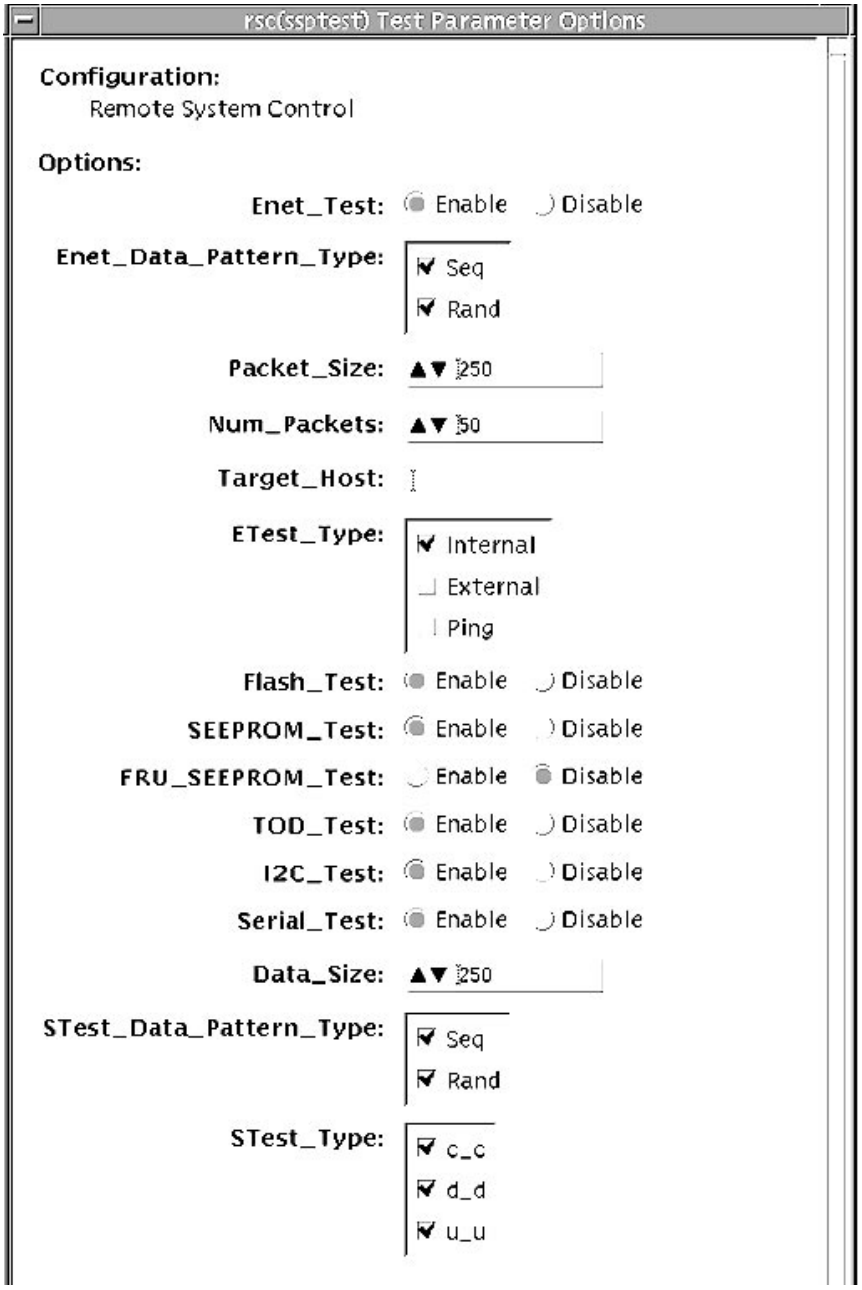

FIGURE 14-1 `ssptest` Test Parameter Options Dialog Box (Top Section)

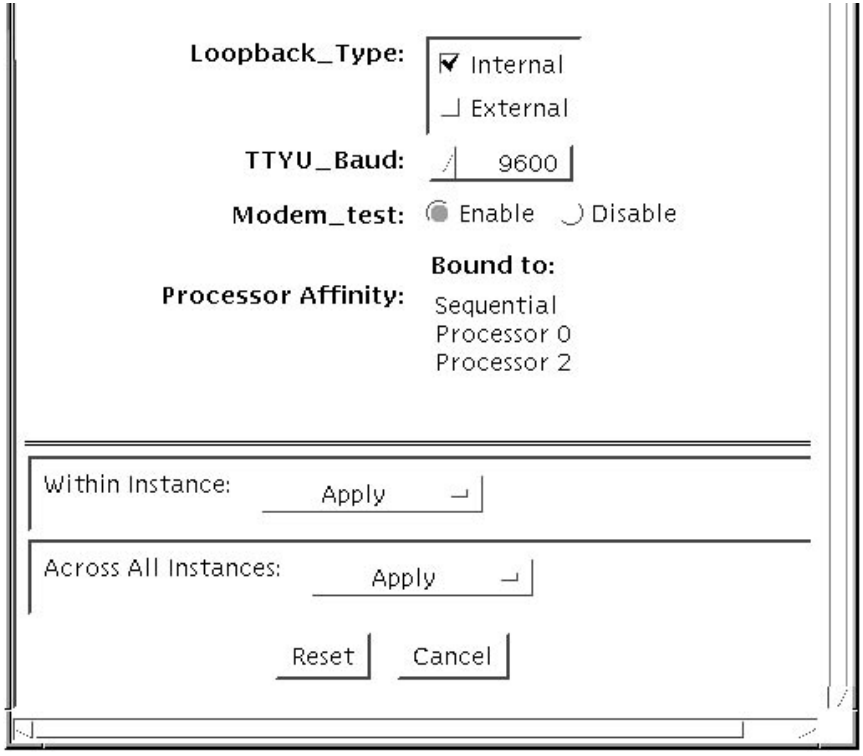

FIGURE 14-2 `ssptest` Test Parameter Options Dialog Box (Bottom Section)

Note – The Configuration field in the `ssptest` Test Parameter Options dialog box displays the which type of hardware (RSC or ALOM) is being tested. For RSC 1.0 and 2.0, *Remote System Control* is displayed. For ALOM, *Advanced Lights-Out Management* is displayed.

TABLE 14-4 `ssptest` Options

`ssptest` **Options**	**Description**
Enet test	Enables or disables RSC or ALOM Ethernet testing.
Data Pattern Type	Selects Sequential, Random, or both types of data patterns.
Packet Size	Defines the size of each data packet to be sent for all tests.
Num Packets	Specifies the number of data packets to send in one test loop.
Target Host	Specifies the IP address of a host to use for the ping test.
Enet Test Type	Selects any or all Internal, External, or ping tests.

TABLE 14-4 `ssptest` Options

`ssptest` **Options**	**Description**
Flash test	Enables or disables the flash checksum test.
SEEPROM test	Enables or disables the SEEPROM checksum test.
FRU SEEPROM test	Enables or disables the FRU SEEPROM checksum test (RSC 2.0 only).
TOD test	Enables or disables the Time Of Day test.
I2C test	Enables or disables the I2C test (RSC 2.0 and ALOM only).
Serial test	Enables or disables the RSC or ALOM serial test.
Data Size	Defines the data size to be sent.
Loopback Type	Selects Internal, External, or both. External requires a loopback plug.
Data Pattern Type	Selects Sequential, Random, or both types of data patterns.
Serial Test Type	Selects serial ports to be tested, u to u, c to c, or d to d.
TTYU_Baud	Select a fixed baud rate or all baud rates for testing the ttyu port. The valid baud rates under TTYU_Baud are: ALL, 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600, 76800, 115200. The default is 9600.
Modem Test	Used to Enable or Disable the RSC PCMCIA modem test (RSC 2.0 only).

`ssptest` Test Modes

`ssptest` supports the following tests as described in the table below.

TABLE 14-5 `ssptest` Supported Test Modes

Test Mode	**Description**
Connection	Reports the status of the RSC or ALOM.
Functional	Tests the RSC's and ALOM's Ethernet, flash, SEEPROM, ToD, I2C, and serial devices. All tests use the internal modes as defaults. The `ssptest` will not run the serial test on `ttyc` if the console has been redirected to the RSC. The `ttu` tests will not run if there is an open login on the `ttyu`.
Exclusive	Tests the RSC's and ALOM's Ethernet, flash, SEEPROM, ToD, I2C, and serial devices. All tests use the internal modes as defaults. The `ssptest` will not run the serial test on `ttyc` if the console has been redirected to the RSC. The `ttu` tests will not run if there is an open login on the `ttyu`.

ssptest Command-Line Syntax

RSC 1.0: **/opt/SUNWvts/bin/ssptest** *standard_arguments* **-o enet=***E/D***,** **epattype=***seq+rand***,esize=***packet_size***,epkts=***number_packets***,target=***IP_address***,** **etest=***I+E+P***,flash=***E/D***,seeprom=***E/D***,serial=***E/D***,sdatsize=***data_size***,** **slb=***I+E***,spattype=***seq+rand***,stest=***u_u+c_c+d_d***,ttyubaud=***baud_rate*|*all*

RSC 2.0: **/opt/SUNWvts/bin/ssptest** *standard_arguments* **-o enet=***E/D***,** **epattype=***seq+rand***,esize=***packet_size***,epkts=***number_packets***,target=***IP_address***,** **etest=***I+E+P***,flash=***E/D***,seeprom=***E/D***,fruseeprom=***E/D***,tod=***E/D***, i2c=** *E/D***,serial=***E*|*D***,sdatsize=***data_size***,slb=***I+E***,spattype=***seq+rand***, stest=** *u_u+c_c+d_d***,ttyubaud=***baud_rate*|*all***,rscmodem=***E/D*

ALOM: **/opt/SUNWvts/bin/ssptest** *standard_arguments* **-o enet=***E/D*, **epattype=***seq+rand*,**esize=***packet_size*,**epkts=***number_packets*,**target=***IP_address*, **etest=***I+E+P*, **flash=***E/D*,**seeprom=***E/D*,**tod=***E/D*,**i2c=***E/D*,**serial=** *E/D*,**sdatsize=***data_size*,**slb=***I*, **spattype**=*seq+rand*, **stest=***d_d*

TABLE 14-6 ssptest Command-Line Syntax

Argument	Description
enet=*enable*|*disable*	Enables or disables RSC or ALOM Ethernet test.
epattype=*seq+rand*	Predefined pattern options used for Enet test.
esize=*packet_size*	Data size for each packet in the Enet test.
epkts=*number_packets*	Number of packets to send for Enet test.
target=*IP_address*	IP address of target system for Enet ping test.
etest=*Internal+External+Ping*	Selects any or all Internal, External, or ping tests.
flash=*enable*|*disable*	Enables or disables RSC or ALOM Flash checksum test.
seeprom=*enable*|*disable*	Enables or disables RSC or ALOM SEEPROM checksum test.
fruseeprom=*E/D (RSC 2.0 ONLY)*	Enables or disables RSC FRU SEEPROM checksum test.
tod=*E/D (RSC 2.0 and ALOM ONLY)*	Enables or disables RSC or ALOM Time of Day test.
i2c=*E/D (RSC 2.0 and ALOM ONLY)*	Enables or disables RSC or ALOM i2c test.
serial=*enable*|*disable*	Enables or disables RSC or ALOM serial test.
sdatsize=*data_size*	Data size for the rsc or alom serial tests.
slb=*Internal+External*	Loopback type. External N/A on ports C and D.

TABLE 14-6 `ssptest` Command-Line Syntax

Argument	Description
`spattype=`*seq+rand*	Predefined pattern options used for RSC or ALOM serial test.
`stest=`*u_u+c_c+d_d*	Defines port and configuration to use for RSC or ALOM serial test.
`ttyu_baud=`*ALL|specific_baud*	Defines baud rates to be used in testing the RSC's console port. The valid baud rates under `ttu_baud` are: ALL, 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600, 76800, 115200. The default is 9600.
`rscmodem=`*E/D*	Enables or disables the RSC PCMCIA modem test.

Note – 64-bit tests are located in the `sparcv9` subdirectory: `/opt/SUNWvts/bin/sparcv9/`*testname, or the relative path to which you installed SunVTS*. If a test is not present in this directory, then it may only be available as a 32-bit test. For more information refer to the "32-Bit and 64-Bit Tests" section of the *SunVTS 5.1 Test Reference Manual* (816-5145-10).